An automatic mesh coarsening technique for three dimensional anisotropic meshes

Youssef Mesri, Hervé Guillard

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

An automatic mesh coarsening technique for three dimensional anisotropic meshes

Youssef Mesri — Hervé Guillard

N° 6344

November 7, 2007

Thème NUM

ISSN 0249-6399

INRIA
SOPHIA ANTIPOLIS

An automatic mesh coarsening technique for three dimensional anisotropic meshes

Youssef Mesri*, Hervé Guillard †

Thème NUM — Systèmes numériques
Projet Smash

Rapport de recherche n° 6344 — November 7, 2007 — 52 pages

Abstract: This work is devoted to the design of a mesh generation technique able to produce a sequence of 3-D coarsened unstructured meshes from an initial anisotropic one. The coarsening algorithm uses an initial mesh and a metric field obtained from an analysis of the natural metric of this initial mesh. First, an initial natural metric (i.e a metric into which each simplicial element of the mesh is equilateral) is produced from the initial anisotropic mesh. Then the eigenvalues of this metric are modified and used together with the initial mesh to produce a coarsened mesh. In this way, the directions of anisotropies of the initial mesh are respected while the mesh spacing can be increased. This procedure can be repeated in order to produce a sequence of semi-coarsened meshes suitable for multigrid acceleration. The efficiency of this procedure is shown on examples of anisotropic meshes involving element aspect ratio as high as 10^5.

Key-words: mesh generation, Coarsening algorithms, anisotropy, Multigrid Algorithms.

* PhD student Smash project
† HDR Smash project

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Méthode de déraffinement anisotrope pour la construction d'une hierarchie de maillages d'une géométrie 3D

Résumé : Ce travail est consacré à la mise au point d'une méthode de génération de séquences de maillages 3D non structurés déraffinés à partir d'un maillage initial fin très anisotrope. Les maillages résultants sont utilisés comme grilles d'un solveur multigrille pour les équations de Navier-Stokes. Le maillage déraffiné est généré à partir du maillage initial et d'une métrique cible obtenue en déraffinant la métrique naturelle calculée sur le maillage fin. On calcule une métrique naturelle sur le maillage à déraffiner (initial) spécifiant la taille de mailles et l'étirement, ensuite on construit une métrique cible basée sur la métrique naturelle et un algorithme de déraffinement. On génère à partir du maillage initial et la métrique cible, un maillage déraffiné. Cette procedure est répétée en fonction du nombre de maillages déraffinés à générer. L'efficacité de l'algorithme de déraffinement automatique du maillage Navies-Stokes est démontrée sur des géométries complexes.

Mots-clés : Génération de maillages, algorithmes de déraffinement, anisotropie, méthodes multigrilles.

1 Introduction

Multigrid algorithms are among the most efficient methods to solve linear and non-linear algebraic problems in computational sciences. However, when non-structured meshes are used, these algorithms suffer from the necessity of constructing a sequence of meshes of different resolution of the same geometry. Two main strategies have been proposed to overcome this problem. The first family of method uses an algebraic interpretation of the problems and relies only on the graph of the matricial problem corresponding to the PDE's that have to be solved. These methods that present several variations [29, 30] are extremely efficient for linear problems, however for non-linear ones, one may prefer the so-called geometric multigrid methods where a sequence of real meshes of different mesh sizes are generated. These geometrical methods can be classified into several families:

- Embedded grids: The simplest manner to build a standard hierarchy of embedded grids consists in first choosing a coarse grid and then to generate the finer level by element subdivision. Unfortunately, this approach has serious limitations. Indeed, as finer and finer grids are built, they become to some extent less and less unstructured since artificial "macro-element" corresponding to the coarse mesh elements are present in the fine triangulation. These macro-elements may influence the computation and show themselves as artificial internal boundaries in the results [33]. Moreover, in an industrial environment, the end user has limited control on the final mesh that comes generally from another department.

- Volume agglomeration: Another idea closely related to algebraic methods is to use the volume agglomeration technique. In this approach, the discretization on the fine grid is associated with control volume defined on a dual mesh. Coarser finite volume grid can then be built by agglomeration of the cells; different algorithms to realize this task are available(see e.g. [12, 13, 14]). A weakness of this approach lies in the difficulty to build second order approximations on the coarse dual grids. Some solutions have however been proposed [31, 32] and the volume agglomeration method is certainly one of the simpler and most efficient ones to deal with non-structured meshes.

- Node-nested method: An alternative to the agglomeration technique consists in building automatically a hierarchy of coarse grids whose elements are not embedded. In 2-D, the first algorithm proposed for this task was in [1] and uses Delaunay re-triangulation associated with a MIS algorithm to delete nodes of the fine triangulation. This algorithm has been followed by 3-D extensions on non-structured isotropic tetrahedral meshes that proved to be very effective for inviscid steady computations [33].

The goal of the present work is to extend these algorithms to the highly anisotropic meshes used for Navier-Stokes computations with Low Reynolds Turbulence modeling and in particular to extend the works done in [2] and [4]. For the meshes we will consider in this work, the maximum aspect ratio of the elements can be very large (typically, more than 10^4 for boundary layers problems) and the multigrid smoothers are quite inefficient for these

cases. The remedy that is known since a long time is to use semi-coarsening in the direction perpendicular to the maximal stretching.

For non-structured grids, this is not an easy task as one as to devise an automatic way to define this direction and then to coarsen the mesh in this direction only. We will realize this task by using the concept of mesh generation controlled by a metric map. Many authors (for example [3], [9]) showed that the definition of a metric field simplifies the generation of adapted and anisotropic meshes : The metrics associated to the Riemannian space specify the size of the mesh and the direction of the stretching and then an adapted and anisotropic mesh in the Euclidean space can be represented as an isotropic and unitary mesh in the Riemannian space.

For semi-coarsening of non-structured meshes, the algorithm that we propose is therefore divided into two steps :

The first one is an analysis step where the current (fine) mesh is analyzed to identify the directions and sizes of stretching. This is done by computing on each element, the metric tensor where this element is equilateral. Then, these tensors are averaged in such a way that on each node of the triangulation, a metric tensor representing the averaged sizes and stretching directions of the elements that contain this node is defined. Then, the eigenvalues of this metric tensor are modified in such a way that the edge lengths of the equilateral element for this tensor is twice the original edge length in the direction of minimal stretching.

The second step is a mesh generation step. We use the fine mesh and the metric field defined on its node as the background mesh and the size specification field as inputs of any existing mesh generation software using the concept of mesh generation governed by metric specifications. In this work, we have used the MTC software [3] but in principle any other mesh generation (Delaunay, frontal, etc) tool can be used for this purpose provided that it can handle anisotropic elements.

The remainder of this paper is organized as follows : Section 2 recall the main concept of a metric map and deal with the problem of metric interpolation. Then in section 3, we describe our directional semi-coarsening algorithm while section 4 gives a quick description of the mesh generator that we have used. Finally, in section 5, we demonstrate the effectiveness of our approach on several anisotropic meshes displaying huge aspect ratio (as large as 10^5).

2 Meshes and Metrics

The concept of mesh generation governed by metric specifications is now of current use in this field [9, 21, 22, 24, 25, 26, 27, 28]. In this section, we review the fundamental properties and definitions of a metric field that will be useful for our applications.
First, we recall the definition of a metric ([9]).

Definition 1:
A Metric (also Tensor) $\mathcal{M}$ in $\mathbb{R}^d$ is a symmetric definite positive real matrix with d the dimension of space. It is then a $d \times d$ invertible matrix, its eigenvalues are real and positive, and its eigenvectors form an orthogonal basis of $\mathbb{R}^d$. such that

$$\mathcal{M} = V(\mathcal{M})^t \Lambda(\mathcal{M}) V(\mathcal{M})$$

where $V(\mathcal{M})$ denotes the orthonormal matrix corresponding to the eigenvectors of $\mathcal{M}$ while $\Lambda(\mathcal{M})$ is the diagonal matrix of its eigenvalues ■

The orthogonalization property of the metric, allows to define a scalar product of two vectors in $\mathbb{R}^d$ with respect to a metric $\mathcal{M}$:

$$(u, v)_{\mathcal{M}} = (u, \mathcal{M}v) = u^t \mathcal{M} v \ \in \ \mathbb{R} \tag{1}$$

we now define, the associated norm of a vector in $\mathbb{R}^d$ with respect to the metric $\mathcal{M}$:

$$\|u\|_{\mathcal{M}} = ((u, u)_{\mathcal{M}})^{1/2} \tag{2}$$

A useful graphical representation of a metric $\mathcal{M}$ associates to it the ellipsoid that represents the unit ball for the scalar product $(.,.)_{\mathcal{M}}$:

$$\mathcal{E}_{\mathcal{M}} = \{u / (u, u)_{\mathcal{M}}^{1/2} = 1\} \tag{3}$$

The equation (3) represents a general ellipsoid form, where the main axis are given by the eigenvectors of $\mathcal{M}$ and the radius of each axis is given by the inverse of the square root of the associated eigenvalues.
More details and definitions can be found in [9].

2.1 Notations

We will use the following notations:

- $V(\mathcal{M})$ an orthonormal matrix corresponding to eigenvectors of a Metric $\mathcal{M}$
- $\lambda_1(\mathcal{M}), \lambda_2(\mathcal{M}), .., \lambda_d(\mathcal{M})$ the eigenvalues corresponding to a metric $\mathcal{M}$.

- $h_1(\mathcal{M}), h_2(\mathcal{M}), .., h_d(\mathcal{M})$ the local size in each direction of the metric defined by: $h_i(\mathcal{M}) = \sqrt{1/\lambda_i(\mathcal{M})}$ for $i = 1, ..., d$.
- For any function f defined on $\mathcal{R}$, we also define the matrix $f(\mathcal{M})$ as the matrix

$$f(\mathcal{M}) = V(\mathcal{M})^t f(\Lambda(\mathcal{M}))V(\mathcal{M})$$

  where $f(\Lambda(\mathcal{M}))$ is the diagonal matrix whose entries are $f(\lambda_i(\mathcal{M}))$

We will suppress the notation $(\mathcal{M})$ when no ambiguity will possible

2.2 Metric associated with a simplicial element

Let $\mathcal{T}_h$ be a tetrahedrization of a polygonal domain Ω defined by a set of nodes $\mathcal{N}$ and a set of simplicial elements (triangles in 2-D, tetrahedron in 3-D) $\mathcal{T}$ where the nodes belong to $\mathcal{N}$. It is quite obvious (and this fact is the basis of the concept of generation of a mesh governed by a metric field, see [9]) that the set of simplicial elements defined a P^0 field of metrics on Ω such that for this metric field each simplex is equilateral : for a given element T, define the set $E(T)$ of all edges of T, $E(T) = \{(k,l) : k \neq l, k, l \in \mathcal{N}, k, l \in T\}$. Then the element metric $\mathcal{M}_T$ on T should satisfy

$$\vec{x}_{kl}^T \mathcal{M} \vec{x}_{kl} = 1 \tag{4}$$

for all edges $(k,l) \in E(T)$ and $\vec{x}_{kl} = \vec{x}_l - \vec{x}_k$ where $\vec{x}_k$ are the coordinates of the node k. This condition gives a system of $1, 3$ or 6 linear algebraic equations (for 1D, 2D and 3D, resp.) for components $(\mathcal{M}_T)_{ij}, i \leq j$ of the metric matrix. The solution of this linear system is given in the following result :

Lemma 2.1 *The element metric matrix M_T can be computed as follows:*

$$M_T = C_M \cdot \left(\sum_{\substack{i,j=1 \\ i<j}}^{d+1} \vec{x}_{ij} \vec{x}_{ij}^T \right)^{-1}, \tag{5}$$

with $C_M = (d+1)/2$. ■

Proof 2.2 *First, let us show that the M_T solving (4) satisfies (5). Indeed, for a solution of (4) we have for all possible edges (i,j), $i < j$ of the simplex:*

$$\vec{x}_{ij}^T M_T \vec{x}_{ij} = 1. \tag{6}$$

We also have the following index-commutativity and transitivity rules:

$$\vec{x}_{ij} = -\vec{x}_{ji} \qquad \textit{and} \qquad \vec{x}_{ik} = \vec{x}_{ij} + \vec{x}_{jk}.$$

From these rules and (6) we have for $j \neq k \neq i$,

$$\begin{aligned} 1 &= \vec{x}_{ij}^T M_T \vec{x}_{ij} = (\vec{x}_{ik} + \vec{x}_{kj})^T M_T (\vec{x}_{ik} + \vec{x}_{kj}) \\ &= \vec{x}_{ik}^T M_T \vec{x}_{ik} + \vec{x}_{kj}^T M_T \vec{x}_{kj} + 2\,\vec{x}_{ik}^T M_T \vec{x}_{kj} = 2 - 2\,\vec{x}_{ki}^T M_T \vec{x}_{kj}. \end{aligned}$$

Hence, by re-indexing

$$\vec{x}_{ij}^T M_T \vec{x}_{ik} = 1/2.$$

In the same way we have for mutually different (i, j, k, m),

$$\vec{x}_{ij}^T M_T \vec{x}_{km} = \vec{x}_{ij}^T M_T (\vec{x}_{kj} + \vec{x}_{jm}) = 1/2 - 1/2 = 0.$$

Now, due to the fact that $\vec{x}_{ij}$ generates $\mathbb{R}^d$ for all possible edges (i, j) of a non-degenerated simplex, the proof consists in showing

$$\left(\sum_{\substack{i,j=1 \\ i<j}}^{d+1} \vec{x}_{ij} \vec{x}_{ij}^T \right) \cdot M_T \cdot \vec{x}_{km} = C_M \cdot \vec{x}_{km}, \tag{7}$$

for all $\vec{x}_{km} \in \mathbb{R}^d$. In fact, it is sufficient to show (7) for only one particular (k, m) and then proceed by cyclic change of node indices. Let us choose $k = 1$, $m = 2$. We get,

$$\begin{aligned} \left(\sum_{\substack{i,j=1 \\ i<j}}^{d+1} \vec{x}_{ij} \vec{x}_{ij}^T \right) \cdot M_T \cdot \vec{x}_{12} &= \sum_{\substack{i,j=1 \\ i<j}}^{d+1} \vec{x}_{ij} \cdot (\vec{x}_{ij}^T M_T \vec{x}_{12}) \\ &= \vec{x}_{12} \cdot (\vec{x}_{12}^T M_T \vec{x}_{12}) + \sum_{\substack{i,j=3 \\ i<j}}^{d+1} \vec{x}_{ij} \cdot (\vec{x}_{ij} M_T \vec{x}_{12}) \\ &\quad + \sum_{j=3}^{d+1} \left[\vec{x}_{1j} \cdot (\vec{x}_{1j}^T M_T \vec{x}_{12}) + \vec{x}_{2j} \cdot (\vec{x}_{2j}^T M_T \vec{x}_{12}) \right] \\ &= \vec{x}_{12} \cdot 1 + \sum_{j=3}^{d+1} [\vec{x}_{1j} \cdot 1/2 - \vec{x}_{j2} \cdot (-1/2)] = \vec{x}_{12} \left[1 + \frac{d-1}{2} \right] = \frac{d+1}{2} \vec{x}_{12}. \end{aligned}$$

Hence, $C_M = (d+1)/2$ and (4) implies (5).

Implication in the other way follows from the regularity (positivity) of the matrix $\sum_{ij} \vec{x}_{ij} \vec{x}_{ij}^T$ and regularity of (4) for any non-degenerate simplex. ■

2.3 Metric associated to the nodes of a tetraedrization

A mesh generation algorithm governed by metric specifications uses a metric field to define at any point of the domain Ω the desired mesh size and stretching directions. However,

this metric field has to be defined in some way. In some cases, it is possible to define this metric field by splitting Ω into several subdomains and defining analytically the metric field over the subdomains. However, in practice, it is more convenient to define this field on an initial mesh, sometimes called the background mesh. This background mesh is only useful for defining at any point of the domain the metric specification and do not have to be confused with the actual mesh that we want to generate. In our coarsening application, the background mesh will obviously be defined by the fine mesh. Using the result of section 2.2, it is clear that we can define a P^0 (constant by element) metric field on the background mesh. However, our coarsening algorithm will remove nodes of the fine triangulation and so thus will delete the elements of the background triangulation. It is thus much more convenient to define a P^1 metric field on the background mesh and thus to define the metric field at the nodes of the background (fine) triangulation. Moreover, to the best of our knowledge, the anisotropic mesh generation softwares currently in use or in development [9, 3] take as input a metric map defined only over the nodes of the background mesh because it is easier to transport P^1 fields during mesh generation process than P^0 fields. We thus have to consider, the problem of defining from a P^0 tensor field a P^1 tensor field that, in some sense has to represent the same field : Given for each element, a metric that defines the length of edges and the stretching directions, we have to define a set of metrics defined on the nodes of the triangulation that will define for any point of the domain approximatively the same lengths and stretching directions. For this, we have considered three different strategies that are described and compared in the following sections.

2.3.1 Interpolation of element metrics to node

A first way is to consider a simple interpolation of the metrics defined on the elements that surround a common node Let us denote $\mathcal{T}(i)$ the set of all elements T which contain a node i, $i \in T$. The nodal metric matrix of node i is simply obtained by:

$$\mathcal{M}_i = (\frac{1}{card(\mathcal{T}(i))} \sum_{T \in \mathcal{T}(i)} \mathcal{M}_T^{-1/2})^{-2} \tag{8}$$

where $\mathcal{M}^{-1/2}$ is a function of the matrix based on the diagonalization of $\mathcal{M} = V.D.V^T$ calculated by taking $\mathcal{M}^{-1/2} = V.D^{-1/2}.V^T$ with a similar definition for $\mathcal{M}^{-2}$. The powers $-1/2$ and -2 have been chosen to give to the expression (8) the meaning of averaging the characteristic mesh size over the elements $T \in \mathcal{T}(i)$ adjacent to i.

2.3.2 Corrected interpolation of element metrics to node

The nodal metric from section 2.3.1 has been constructed on the element metrics, and as such should take into account the local shape and direction of anisotropies of adjacent mesh elements. However, numerical experiments show that it does not give, in some cases, quite exact information on the average aspect ratio of the adjacent elements and that it oversizes the smallest characteristic mesh size (eg. the boundary layer thickness). This situation

happens, for example, on a curved boundary layer with a high aspect order (order 10^4 or more).
For such a case, we propose a correction of the eigenvalues of the nodal metric matrix M_i computed in (8). For each metric matrix M, let us denote its diagonal matrix by D with eigenvalues ordered in an ascending order on the diagonal and an orthonormal matrix of corresponding eigenvectors by V,

$$M = V.D.V^T$$

With reference to expression (8) let us compute a modified diagonal matrix $\bar{\mathcal{D}}_i$ as follows

$$\bar{\mathcal{D}}_i = (\frac{1}{card(\mathcal{T}(i))} \sum_{T \in \mathcal{T}(i)} \mathcal{D}_T^{-1/2})^{-2} \tag{9}$$

The final nodal metric is thus given by

$$\mathcal{M}_i = V_i.\bar{\mathcal{D}}_i.V_i^T \tag{10}$$

where V_i is the orthogonal matrix of eigenvectors of the original matrix M_i used in formula (8). This procedure thus replaces the eigenvalues of $\mathcal{M}_i$ by the corrected eigenvalues given by (9) that tries to have a better estimate of the aspect ratio of the elements at node i.

2.3.3 Interpolation as computing geodesics

The problem of interpolating tensor fields has recently received attention in the area of image processing [10]. We describe in the following two sections, two different strategies to define a nodal metric field coming from this research area. Let us again consider $\mathcal{T}(i)$ the set of all elements T which contain a node i, $i \in T$. On each element T is defined by the results of section 2.2 a unique symmetric positive definite tensor (SPD) M_T. We are looking for a unique metric M_i that represents in the best possible way this set of tensors. This problem can be solved as a minimization problem provided we are able to define a concept of distance between tensors. For this we have to recall some definitions and results.

First, let us consider the definition of the geodesic distance between two symmetric positive definite tensors. In the sequel, we will denote $Sym(n)$ the set of symmetric real n-matrices and $Sym_*^+(n)$, the set of SPD real n-matrices :

Definition 1 :
The invariant distance between $M, N \in Sym_*^+(n)$ is defined by:

$$d(M,N) = dist(Id, M^{-1/2}NM^{-1/2}) = \mathcal{N}(M^{-1/2}NM^{-1/2})$$

and

$$\mathcal{N}(M)) = tr(log(D)^2).$$

where D is the eigenvalue matrix associated to M.

Another way to determine the invariant distance is through the Riemannian metric on the space of SPD tensors.

Definition 2 :
if W_1 and W_2 are two tangent vectors at M, we define the scalar product at M by:

$$(W_1, W_2)_M = (M^{-1/2} W_1 M^{-1/2}, M^{-1/2} W_2 M^{-1/2}) = Tr(M^{-1/2} W_1 M^{-1} W_2 M^{-1/2})$$

Definition 3 :
The Riemannian exponential map associated to an invariant metric is defined by:

$$exp_M(W) = M^{1/2} exp(M^{-1/2} W M^{-1/2}) M^{1/2}$$

this diffeomorphism is global, and we can uniquely define the inverse mapping by:

$$log_M(W) = M^{1/2} log(M^{-1/2} W M^{-1/2}) M^{1/2}$$

Remark: in appendix B, we describe in details the properties of Riemannian space of symmetric positive definite tensors as well as the definitions of this section.

Intrinsic tensors mean as minimization algorithm:

With the previously defined concept of geodesic distance between SPD tensors, we can define the nodal metric M_i as the intrinsic mean of the set of metrics $\{\mathcal{M}_T; T \in \mathcal{T}(i)\}$ [16]. We recall that the intrinsic mean of a random variable in an arbitrary metric space is defined in the sense of Fréchet as the point that minimizes the expected value of the sum-of-squared distance function : Consider a set of point $A = \{x1, \cdots, x_n\}$ on a Riemannian space $\mathcal{E}$. Then the intrinsic mean of points in A is defined as a minimum of the function j_A, that is,

$$\mu = \text{argmin}_{x \in \mathcal{E}} j_A(x) \tag{11}$$

where

$$j_A(x) = \frac{1}{2n} \sum_{i=1}^{n} d(\mu, x_i)^2$$

and d is the geodesic distance on $\mathcal{E}$.

The properties of the intrinsic mean have been studied by Karcher [17]. Since the mean is given by the minimization problem (11), we must verify that such a minimum exists and is unique. Karcher shows that for a manifold with non-positive sectional curvature the mean is uniquely defined. In fact, the SPD space does have non-positive sectional curvature, and,

thus, the mean is uniquely defined. In [18] a gradient descent algorithm to compute this mean is given. The gradient of j_A is given by

$$\nabla j_A(x) = -\frac{1}{n} \sum_{i=1,n} Log_x(x_i)$$

and thus the intrinsic mean of a set of tensors can be computed by the following gradient descent algorithm :

Algorithm 3: Intrinsic Mean of Tensors
Input: $M_1, \cdots, M_n \in SPD$ space
Output: $R \in SPD$ space, the intrinsic mean

$R_0 = M_1$

Do
$X_i = \frac{1}{n} \sum_{k=1,n} Log_{R_i}(M_k)$
$R_{i+1} = Exp_{R_i}(X_i)$

While $(X_i, X_i)_{R_i} > \epsilon$

2.3.4 Interpolation in Log-Euclidean space

One of the difficulties in dealing with metrics is that this space is not a vector space for the usual matrix and scalar multiplication operations. The purpose of this section is the use of a new family of matrix operations on the space $Sym_*^+(n)$ that will give to this space a vector space structure [10]. These operations that generalize the usual matrix and scalar multiplications are interesting in term of their mathematics properties such as the similarity-invariant, invariant by multiplication, inversion and by scaling in the domain of matrix logarithms.
Definitions Let $M_1, M_2 \in Sym_*^+(n)$ and $\lambda \in \mathcal{R}$. a logarithmic product of M_1, M_2 is defined as follow:

$$M_1 \odot M_2 := exp(log(M_1) + log(M_2)) \tag{12}$$

and a logarithmic scalar multiplication $\otimes$ is defined by:

$$\lambda \otimes M_1 := exp(\lambda.log(M_1)) = M_1^\lambda \tag{13}$$

Proposition 1. $(Sym_*^+, \odot)$ is a group. The neutral element is the usual identity matrix, and the group inverse of an SPD matrix is its inverse in the matrix sense. Moreover, whenever two SPD matrices commute in the matrix sense, then the logarithmic multiplication is equal to their matrix product. Moreover, the multiplication is commutative. ■

Proposition 2.

$$exp : (Sym, +) :\longrightarrow (Sym_*^+, \odot) \tag{14}$$

is Lie group isomorphism. ■

In particular, one-parameter subgroups of $Sym_*^+(n)$ are obtained by taking the matrix exponential of those of $Sym(n)$, which are simply of the form $(t, V)_{t\in\mathbb{R}}$ where $V \in Sym(n)$. As consequence, the Lie group exponential in $Sym_*^+(n)$ is given by the classical matrix exponential on the Lie Algebra $Sym(n)$.
A structure of Lie group is given to the space of SPD matrices. The new multiplication used, i.e the logarithmic multiplication, generalizes the matrix multiplication when two SPD matrices do not commute in the matrix sense.

Proposition 3.: $(Sym_*^+(n), \odot, \otimes)$ has a vector space structure. In the sequel, we will denote the space $(Sym_*^+(n)$ endowed with these two operations the Log-Euclidean vector space of metrics.

A distance between two SPD matrices in Log-Euclidean space can be defined by :

$$d(M_1, M_2) = ||log(M_2) - log(M_1)||$$

where $||.||$ denotes here the metric norm. ■

Another important result in Log-Euclidean space, which concerns interpolation of metrics in Log-Euclidean space is given in the following proposition :

Proposition 4.: Let $(M_i)_1^N$ be a finite number of SPD matrices. Then their Log-Euclidean Fréchet mean exists and is unique. It is given by:

$$M_{LE}(M_1, \cdots, M_N) = exp(\frac{1}{N}\sum_{i=1}^{N} log(M_i))$$

■

Corollary 1.: The Log-Euclidean mean is similarity-invariant, invariant by group multiplication and inversion, and is exponential-invariant (i.e invariant with respect to scaling in the domain of logarithms) ■

The previous propositions (from 1 to 4) are proved in [10].

2.3.5 Comparison between the previous strategies

We compare in this section, the three ways to define the nodal metric field described in the previous sections. In the sequel, these three strategies will be labeled respectively interpo (section 2.3.2), geodesic (section 2.3.3) and Log (section 2.3.4).
Recalling that our main problem concerns highly stretched meshes used in boundary layers, we first compare these three strategies on a 2-D model boundary layer mesh shown in figure

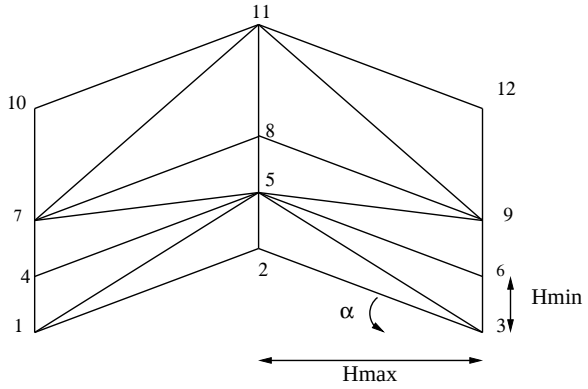


Figure 1: 2-D model boundary layer mesh

1. Curved boundary layers are often the regions where the distance between the tensors defined on the elements of the fine mesh is largest. To mimic this behavior, we will vary on figure 1 the angle α. $\alpha = 0$ thus represents a plane surface and large α angles will represent highly curved surfaces. The ratio h_{Max}/h_{min} on figure 1 define the stretching of the mesh.

We first compare the three strategies on a plane boundary layer ($\alpha = 0$) with respectively a stretching ratio of $h_{Max}/h_{min} = 100$ and $h_{Max}/h_{min} = 1000$. Thus on these case, the eigenvectors of the metric matrices defined by triangles are almost identical and these matrices differ principally by their eigenvalues. The results are shown on figure 2.
For these two stretching ratios, the results given by the three interpolation strategies are virtually identical. The interp procedure producing more isotropic results than the other two ones. However, on these figures, in order to visualize the results, the units on the y-axis have been multiplied by a factor proportional to the stretching ratio (if the ellipsis should have been represented in their real size, they would have appeared as a single lines). Thus the difference between the three metrics are over-represented. Finally also note that the results for the two stretching ratio can be deduced from each others by a simple scaling of the y-axis.

We now consider the case of a curved boundary layer with $\alpha = 20^0$. The eigenvectors of the metric matrices on each individual triangles are thus in this case quite different and the interpolation of these matrices is difficult. The first point to note is that, although the existence of the minimum in expression (11) is theoretically proven ([17]), for these cases and for these stretching ratio, the gradient descent algorithm of section 2.3.3 does not converge anymore. In the sequel, we will thus only compare the results given by the interpo and Log procedures. The ellipsis describing the unit balls for the two strategies are shown in

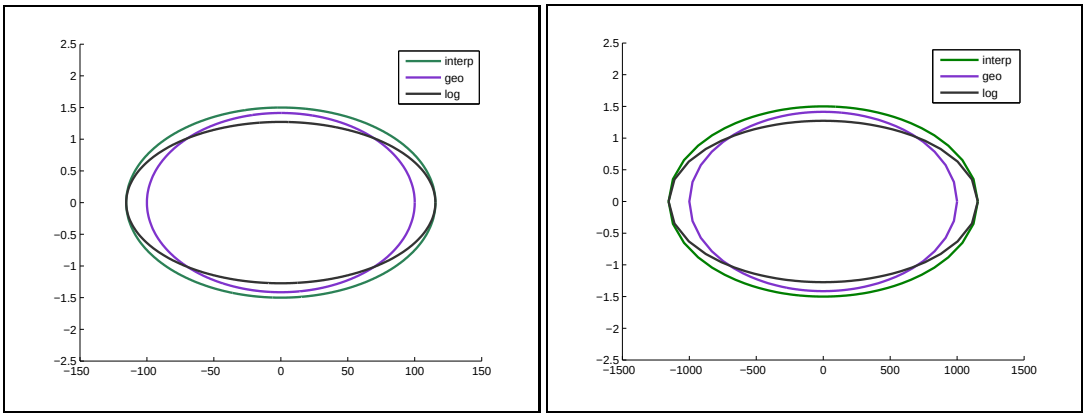


Figure 2: Comparison of the unit ball for the metric defined on the mesh of figure 1. Left : Stretching ratio of 100; Right : Stretching ratio of 1000. Note that the units on the y-axis are scaled with a factor proportional to the stretching ratio in order to be able to visualize the unit balls

figure 3. We emphasize that the results are much more closer than they seem on these figures given the large scaling of the y-axis performed to visualize the results. Anyway, the interp procedure has a more pronounced effect of "balancing" the eigenvectors and the metric appear to be stretched in the x-direction while this effect is less pronounced for the Log procedure (Note that on these figures, if the metric corresponding to the log procedure appear actually to be isotropic or stretched in the y-direction, this is just an optical effect due to the scaling of the y-axis).

At this stage, the results obtained with the interp and Log procedures do not allow to prefer one of them over the other. In our application, the definition of the nodal metric field is used to produce coarsened meshes. We thus use the two strategies interp and Log to real 3-D mesh of an airfoil described in section 5.2. This mesh is depicted in figure 8. The initial fine mesh contains 67866 nodes and stretched elements are present in the boundary layer : more than 5 % of the mesh have an aspect ratio larger than 150. Table 2.3.5 compares the number of nodes and coarsening ratio [1] obtained by the two procedures. It is seen that the definition (9) of section 2.3.2 (the interp procedure) gives a much more aggressive coarsening than the use of the log procedure : The size of the **fourth** mesh obtained with the log procedure is roughly equivalent to the size of the **third** mesh obtained with the interp procedure. Since the size of the meshes is an important aspect for the complexity of

[1] ratio of the number of nodes in two successive fine and coarse triangulations : in 3-D, this number should be equal to 2 in the semi-coarsening case and to 8 for an isotropic coarsening

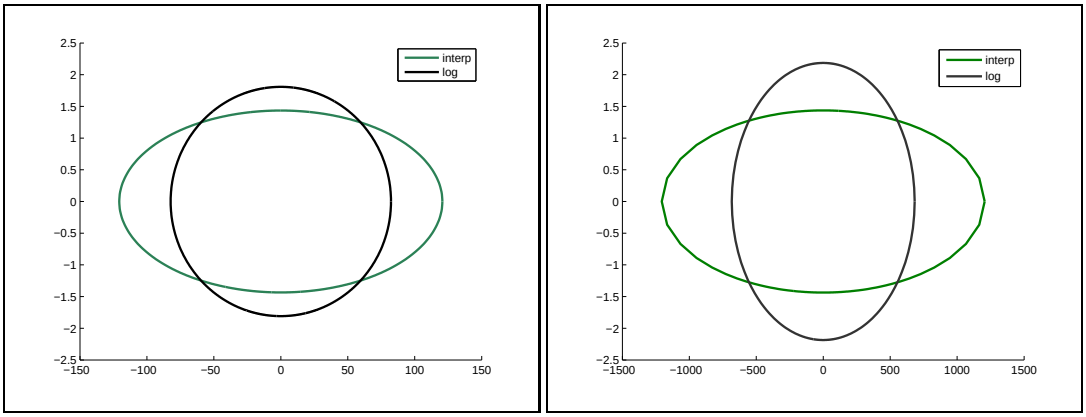


Figure 3: Comparison of the unit ball for the metric defined on the mesh of figure 1 with $\alpha = 20^0$. Left : Stretching ratio of 100; Right : Stretching ratio of 1000. Note that the units on the y-axis are scaled with a factor proportional to the stretching ratio in order to be able to visualize the unit balls

	interp		log	
	# of nodes	coarsening ratio	# of nodes	coarsening ratio
initial mesh	67866			
first coarsened mesh	30961	2.19	39010	1.74
second coarsened mesh	15123	1.99	22264	1.75
third coarsened mesh	8507	1.77	13246	1.68
fourth coarsened mesh	5213	1.63	8323	1.59

Table 1: Comparison of interp and log methods on the M6 mesh.

a multigrid algorithm, we will prefer the interp procedure and this is this procedure that will be used in the sequel.

3 Anisotropic coarsening algorithm

3.1 Original anisotropic coarsening algorithm

In this section we describe briefly the main steps of an automatic coarsening algorithm that takes into account the mesh anisotropy. This algorithm have been designed in a previous study by Ales Janka and Hervé Guillard. For more details see [2] .

This algorithm follows the following steps :

• Generate on each node of the finest mesh an initial nodal metric that reflects the size and stretching of elements belonging to this mesh.

• Modify the initial metric to establish a corresponding coarsened mesh metric. This is done by modifying the eigenvalue λ_i associated to the metric, which will modify the mesh size in the desired direction which is the corresponding eigenvector V_i.

• Provide the background (fine) mesh and the desired metric field to a mesh generation tool using metric specifications.

Algorithm 1: Anisotropic semi-coarsening algorithm
Input: $\lambda_1, \cdots, \lambda_d$ eigenvalues of a $\mathcal{M}_k$
Output: Update $\lambda_1, \cdots, \lambda_d$ coarsened eigenvalues

For each $i \in \mathcal{N}$

Set: $h_k^i = (\lambda_k^i)^{-1/2}$, $k = 1, \ldots, d$, $h_1^i \leq \cdots \leq h_d^i$
and $h_0 = C_{cf}.h_1$.

$h_k^i = \max\big(h_k^i, \min\left(C_{cf} \cdot h_k^i, h_{k-1}^i\right)\big)$.

Define new eigenvalues $\lambda_k \leftarrow h_k^{-2}$.

End for.

3.2 Modified anisotropic coarsening algorithm

The previous anisotropic coarsening algorithm produces a semi-coarsening in the boundary layer region and a total coarsening in the isotropic region. However, as the different meshes are progressively coarsened, it may happen that this algorithm produces an extremely rapid change in the metric specifications. To understand this point, consider Figure 4. It represents on the left, a configuration of a mesh zone between boundary layer and isotropic mesh zone. On the right, we have shown a representation of the metric at the nodes i and j respectively. Observe that the x-radius of the two ellipsis are equal. The previous algorithm will identify node i as being in the isotropic region and consequently, will ask for a doubling of the radius of the ellipsis in the x and y direction. However, point j will be identified as being in the boundary layer and therefore, the metric will be modified only in the y-direction. The metric specification for the new coarsened meshes is therefore represented in figure 5. It is clear that we have a conflict between the two opposite requirements given by these metrics. This situation has been studied in other context in [8] and mesh generation algorithms have difficulties to handle these cases. A solution is to allow a smooth evolution of the metric field. There are many different ways to obtain a regular metric map. For instance, one can use

smoothing of the metric field by averaging the different metric on an enlarged stencil around a node using the technique described in section 2.3. Here, we simply suggest an improved algorithm that allows to coarsen smoothly the intermediate region between anisotropic and isotropic mesh regions.

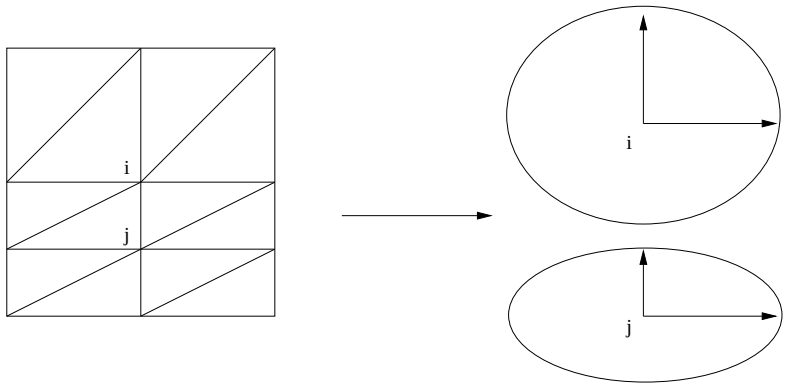


Figure 4: metrics at interface points between two different mesh anisotropy zones

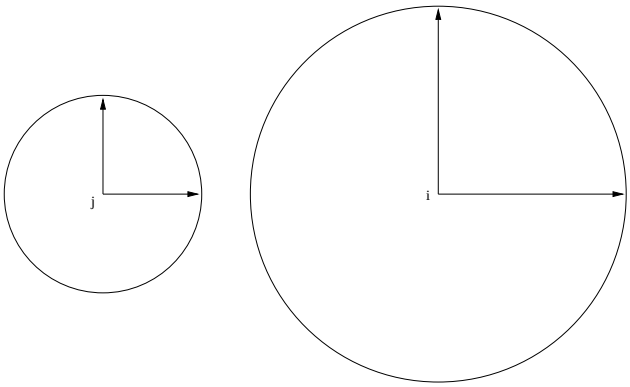


Figure 5: metrics at interface points after application of the coarsening algorithm

This algorithm takes into account the specification of the nodal metric on the neighbors of a given node and do not allow a fast variation of the metric specifications. This algorithm is

the following :

Algorithm 2: Modified anisotropic coarsening algorithm (MACA)
Input: $\lambda_1, \cdots, \lambda_d$ eigenvalues of a $\mathcal{M}_k$
Output: Update $\lambda_1, \cdots, \lambda_d$ coarsened eigenvalues

For each $i \in \mathcal{N}$

Set: $h_k^i = (\lambda_k^i)^{-1/2}$, $k = 1, \ldots, d$, $h_1^i \leq \cdots \leq h_d^i$.

Set $h_0 = C_{CF} \cdot h_1$.

Store the initial size: $h^i_{k,\text{old}} = h_k^i$.

// compare the current node size specifications with its neighbors

if $\max\big(h_d^i, \min\left(C_{cf} \cdot h_d^i, h_{d-1}^i\right)\big) = C_{cf} \cdot h_d^i$

and $\exists j_0 < i \in V(i)$ such as
$h_1^{j_0} < h_1^i$. then $h_k^i = h_{k,old}^i, k = 1, \ldots d$

Define new eigenvalues $\lambda_k \leftarrow h_k^{-2}$.

End for

4 The MTC mesh generator

The coarsening algorithm developed in this work can be used with any mesh generator using the concept of generation governed by a metric map (or alternatively by stretching directions). These mesh generators can use the Delaunay principle, advancing fronts or layers or any other kind of mesh generation method. However, in practice, we have found that for three-dimensional geometries, the number of mesh generation softwares able to handle the extremely anisotropic meshes typical of CFD applications is extremely limited. Thus in this section, we give a short description of the MTC mesh-generation tool that we have used in this work. MTC is being developed by Thierry Coupez and his Team at Ecole des Mines de Paris, Centre de Mise en Forme des Materiaux, Sophia Antipolis. It is based on the idea to improve iteratively, an initial unsatisfactory mesh by local improvements. The general algorithm can be expressed as follows, for further details see [3].

4.1 Meshing and re-meshing processes in MTC

MTC mesh generator re-mesh the initial mesh iteratively by a local mesh optimization technique. The mesh optimization technique consists in local re-meshing of cavities formed by small clusters of elements in order to increase the "quality" of the elements of the cluster.

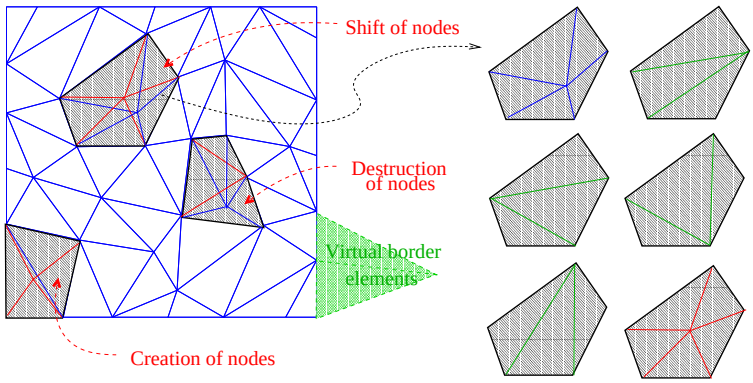


Figure 6: Local mesh optimization process in MTC

In the re-meshing process, two principles are enforced :

- **Minimal volume**, which assures the conformity of the mesh, with no overlaps of elements: let $\mathcal{T}_i(C)$ denote the i-th set of elements T filling the local cavity. Following the minimum volume principle we choose as an optimal (possibly not unique) re-triangulation of the cavity the one satisfying

$$\sum_{T \in \mathcal{T}_i(C)} |(Volume)(T))| \rightarrow \min, \tag{15}$$

  where the minimization is done over a small set $i = \{1, \ldots, I\}$ of possible triangulations $\mathcal{T}_i(C)$ of elements (Fig. 6 right) connecting the nodes on the border of the cavity, or other nodes like the cavity barycenter, with all boundary faces.

- **The geometrical quality** $Q(T)$, which is evaluated for each element. If the minimizer of (15) is not unique, this criterion picks among all admissible cavity re-triangulations the one improving the geometrical quality of the mesh by improving the quality of the worst element of the triangulation.

While the former criterion assures the conformity of the mesh, if the initial mesh was conform, the latter handles improvements of element shape, size, connectivity, etc., depending on the quality function $Q(T)$. Usually, the quality function $Q(T)$ is a function of the geometry of the element T and the prescribed background metric, which give together a measure for the element size and the element form (aspect ratio).

4.2 Definition of the quality function

Let $C_n = \{T_1, \cdots, T_n\} \subset T_h$ be a set of n elements. We define the quality of this set with respect to a given metric $M \in \mathbb{R}^{d\times d}$ as a real *n-vector*

$$Q_n = \{q(T_1), \cdots, q(T_n)\}$$

The quality $q(T)$ of an element T of C_n, measured in the metric M, is defined as the product of two factors :

$$q(T) = Q_F(T) \cdot Q_S(T) \tag{16}$$

The first one $Q_F(T)$ control the shape of the element T. The best possible element being the equilateral element (in the metric M). It is defined by :

$$Q_F(T) = C_0 \frac{(Volume)(T)_{M(T)}}{h^d_{M(T)}} \tag{17}$$

where

- d is the space dimension,
- $(Volume)(T)_{M(T)}$ is the volume of T measured in the metric space, and is given by:

  $$(Volume)(T)_{M(T)} \;=\; (Volume)(T)\sqrt{det(M(T))}.$$

  The matrix $M(T)$ is obtained by averaging the nodal metric matrices on nodes of the element T,

  $$M(T) = \left(\frac{1}{d+1}\sum_{i\in T} M_i^{-1/2}\right)^{-2},$$

  and correct its eigenvalues by the same process as in Section 2.3.2.
- $h_{M(T)}$ is the average of lengths of edges of T measured by the metric $M(T)$,

  $$h_{M(T)} = (\frac{2}{d(d+1)} \sum_{(i,j)\in T} (M(T)(\vec{x}_j - \vec{x}_i), (\vec{x}_j - \vec{x}_i)))^{1/2} \tag{18}$$

- in (17) C_0 is such that 17 is equal to 1 when T is equilateral in the metric $M(T)$.

The second factor $Q_S(T)$ controls the *size* of the element in the metric M. Its definition is given by :

$$Q_S(T) = min(\frac{1}{h_{M(T)}}, h_{M(T)})^d \tag{19}$$

where $\vec{x}_i \in \mathbb{R}^d$ are the coordinates of the node i.

With these definition of the quality of a set of elements, we just need now a way to compare two different sets in order to pick the "best" one. This is done by defining an ordering relation " $<''$ between sets of elements as follows : Let C_i and C_j be two set of elements. Denote by Q_i and Q_j the qualities of the two set C_i and C_j. Set $\overline{C}_i = C_i$ and $\overline{C}_j = C_j$. Let us say that $Q_i < Q_j$ when for the worst-quality elements $T_i \in \overline{C}_i$ and $T_j \in \overline{C}_j$, $q(T_i) = min(Q_i)$, $q(T_j) = min(Q_j)$, we have $q(T_i) < q(T_j)$. If $q(T_i) = q(T_j)$, set $\overline{C}_i \leftarrow C_i \backslash T_i, \overline{C}_j \leftarrow C_j \backslash T_j$ and repeat the comparison.

4.3 Algorithm (MTC iteration)

With the concept previously defined, the mesh generation in MTC will proceed by successive improvements of given mesh. Let us denote T_h, E_h, N_h, respectively the sets of all mesh elements, edges and nodes. Repeat for different cavities $C_0 \in T_h$ composed by a group of adjacent elements obtained as the nearest neighborhood of a node $n \in N_h$ or of an edge connecting 2 nodes $(n_1, n_2) \in E_h$, $n_1, n_2 \in N_h$.

1. Denote $N_0 \subset N_h$ and $E_0 \subset E_h$ respectively, the set of all nodes and all edges of the cavity C_0.

2. Denote $\partial E \subset E_0$ the set of all edges belonging to the border of the cavity C_0.

3. Evaluate the quality Q_0 of the set C_0 of elements $T \in C_0$ by a given function, e.g. 16 and 18.

4. For each $n \in N_0$ do:

- Connect the node n to each edge $e \in \partial E_0, n \notin e$ to get a set of elements $C_n = \{T_e, T_e = (e, n), \text{attempting to re triangulate the cavity } C_0\}$.

- Evaluate the quality Q_n of the set C_n by a given function, e.g. 16.

- If $Q_0 < Q_n$ (in the sense of Definition 4.2) then set $C_0 \leftarrow C_n, Q_0 \leftarrow Q_n$.

Until stagnation of the changes applied to the mesh.

4.4 Modified Quality function

In our application, we have found useful to re-define slightly some component of the MTC algorithm. This section details these modifications. First, we have re-defined the functions

$Q_S(T)$ and $Q_F(T)$ in expression (16) in the following way. We take

$$Q_S(T) = \min_{(i,j)\in T} \left(h_{ij}^{M(T)}), \frac{1}{h_{ij}^{M(T)}} \right)^d ,$$

instead of

$$Q_S^{orig}(T) = \min \left(h_{M(T)}, \frac{1}{h_{M(T)}} \right)^d ,$$

This new definition avoid taking averages, because the mesh lengths may be very different on an element due to the anisotropy of the mesh.

The measure of the shape quality remains the same as in the original MTC code,

$$Q_F(T) = C_0 \frac{(Volume)_{M(T)}(T)}{h_{M(T)}})^d,$$

Finally, we have found useful to add to the measure of the quality of an element (16) an additional function $Q_V(T)$ to have a better control on the element characteristic volume. Thus our definition of the quality of an element is now

$$Q(T) = Q_S(T) \cdot Q_F(T) \cdot Q_V(T)$$

This increase the sensitivity of $Q(T)$ to the non-respect of prescribed aspect ratio. The definition of $Q_V(T)$ that we use is

$$Q_V(T) = \min \left(\frac{1}{(Volume)_{M(T)}(T)}, (Volume)_{M(T)}(T) \right).$$

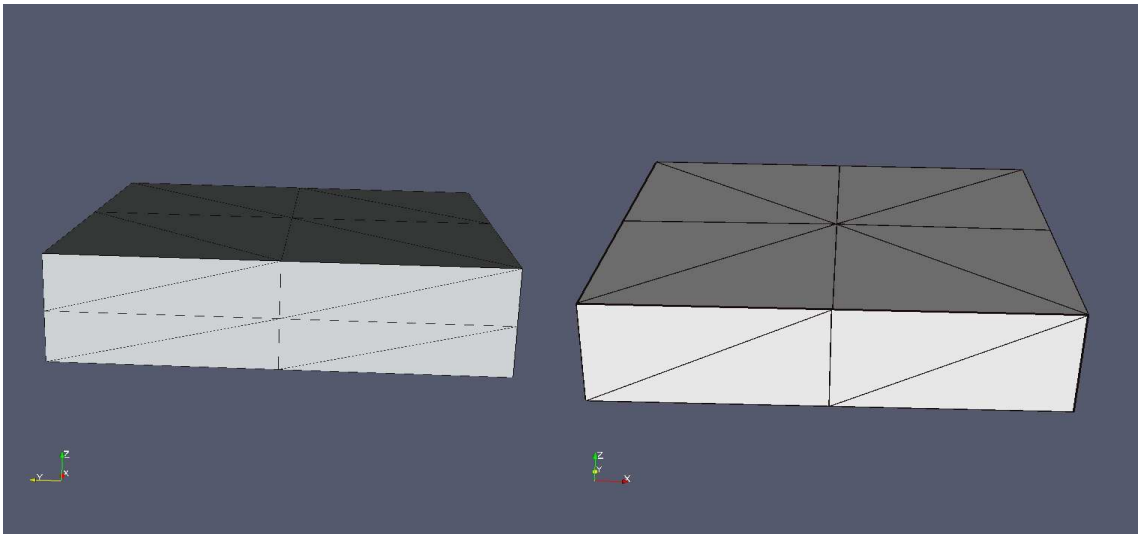
5 Applications

5.1 Coarsening synthetic mesh

The coarsening process and development detailed in the previous sections, is applied here on a model problem in order to show how it works. Then in the following section, we will apply the semi-coarsening tool to industrial meshes with a very large aspect ratio.

Our model problem is a regular hexaedral domain Ω divided into 8 hexaedrons whose volumes are $V = \delta x \times \delta y \times \delta z$, we take $\delta x = \delta y$ and $\delta z << \delta x$. In the example given below, $\delta z = 10^{-3}\delta x$ but we have checked that the results does not depend on this ratio and that the results are unchanged even with $\delta z = 10^{-6}\delta x$. Each hexaedron is divided into two prisms, each being subdivided into three tetrahedra. Figure 5.1 (a) shows this tetrahedrization that contains 27 nodes and is composed of three node layers in the z direction On each of these 27 node, the eigenvectors of the corresponding metric are approximately the three x, y, z axis with corresponding eigenvalues respectively equal to $(1/\delta x)^2, (1/\delta y)^2, (1/\delta z)^2$.

The coarsening algorithm will change this metric for another one with the same eigenvectors but with eigenvalues corresponding to the lenght $\delta x, \delta y, 2\delta z$ then the mesh generator will try to construct a mesh corresponding to this wished metric. The result is shown on figure [5.1 (b)]. As expected, this mesh contains with 18 nodes and the coarsening process has eliminated the intermediate node layer thus performing a semi-coarsening by a factor 2 in the z direction.



(a)Baseline mesh (b)Coarsened mesh

Figure 7: Semi-coarsening of a model boundary layer mesh. Note that the y-axis is scaled in order to be able to visualize the result.

5.2 M6 wing test case

Our second example is an industrial mesh of an $M6$ airfoil. This mesh shown in figure 8 contains 67866 nodes with highly stretched elements in the boundary layer. Figure 9 shows an enlarged view of the mesh near the airfoil and the symmetry axis where stretched boundary layer elements are visible. Another way to measure the anisotropy of the mesh is to perform a classification of the nodes according to their aspect ratio. Since semi-coarsening for MG method is usually done by doubling the mesh size by a factor 2 in the direction perpendicular to the direction of maximal stretching, this classification is done in power of 2 : The first class contains vertices having an aspect ratio between $ratio\_max$ and $ratio\_max/2$, the second one includes those having an aspect ratio between $ratio\_max/2$ and $ratio\_max/2^2$ and so on. The last class contains vertices having an aspect ratio around 1.

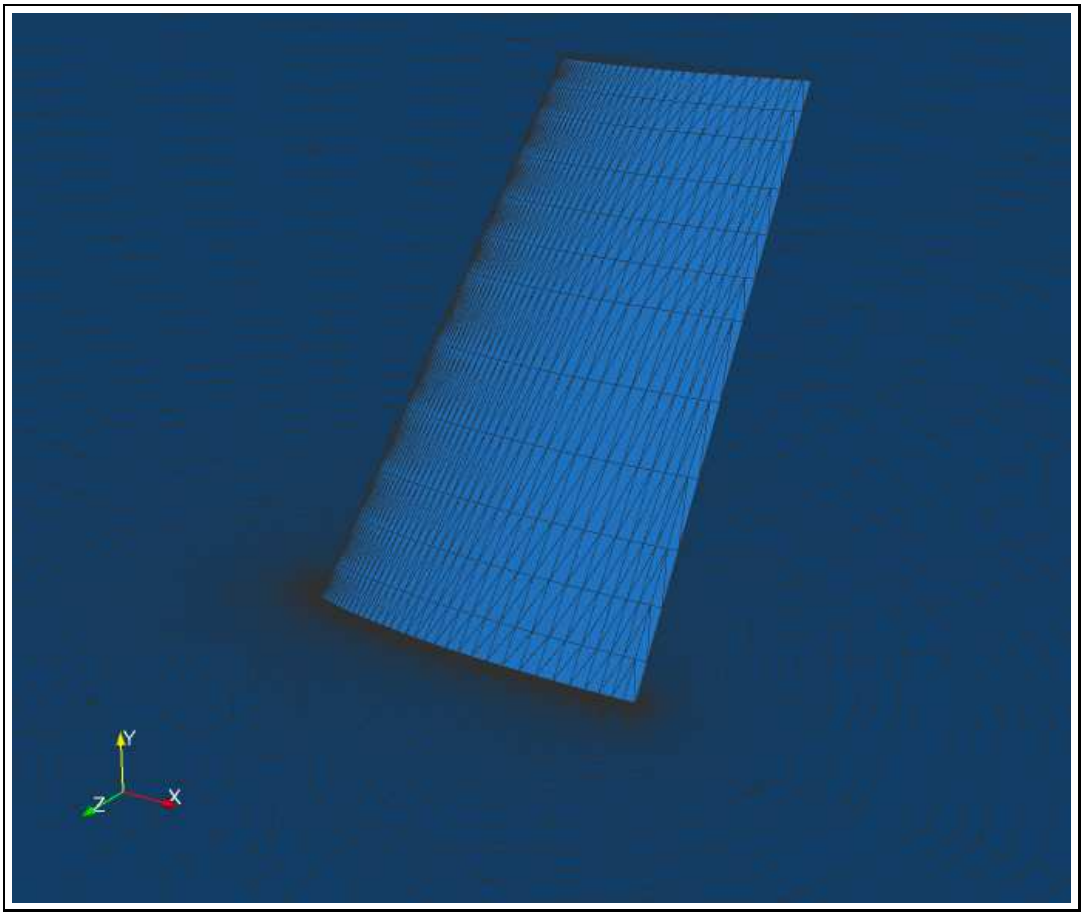


Figure 8: M6 wing : initial finest mesh

Figure 11 (blue line) displays the histogram corresponding to this node ordering. The y-coordinate corresponds to the number of nodes in each class divided by the total of nodes. This histogram. shows that more than 5 % of the mesh has an aspect ratio larger than 150.

Using the previously described coarsening algorithm, we have generated four coarsened meshes, the number of nodes obtained for these meshes is given in Table 2. It is seen that the coarsening ratio is as expected of the order of 2. The coarsening ratio decreases as successive application of the coarsening algorithm progresses. This is due to geometric constraints : As more and more nodes are eliminated, it becomes more and more difficult to respect the geometry. Neverthless, the result is very satifactory. Figure 10 shows the same enlarged

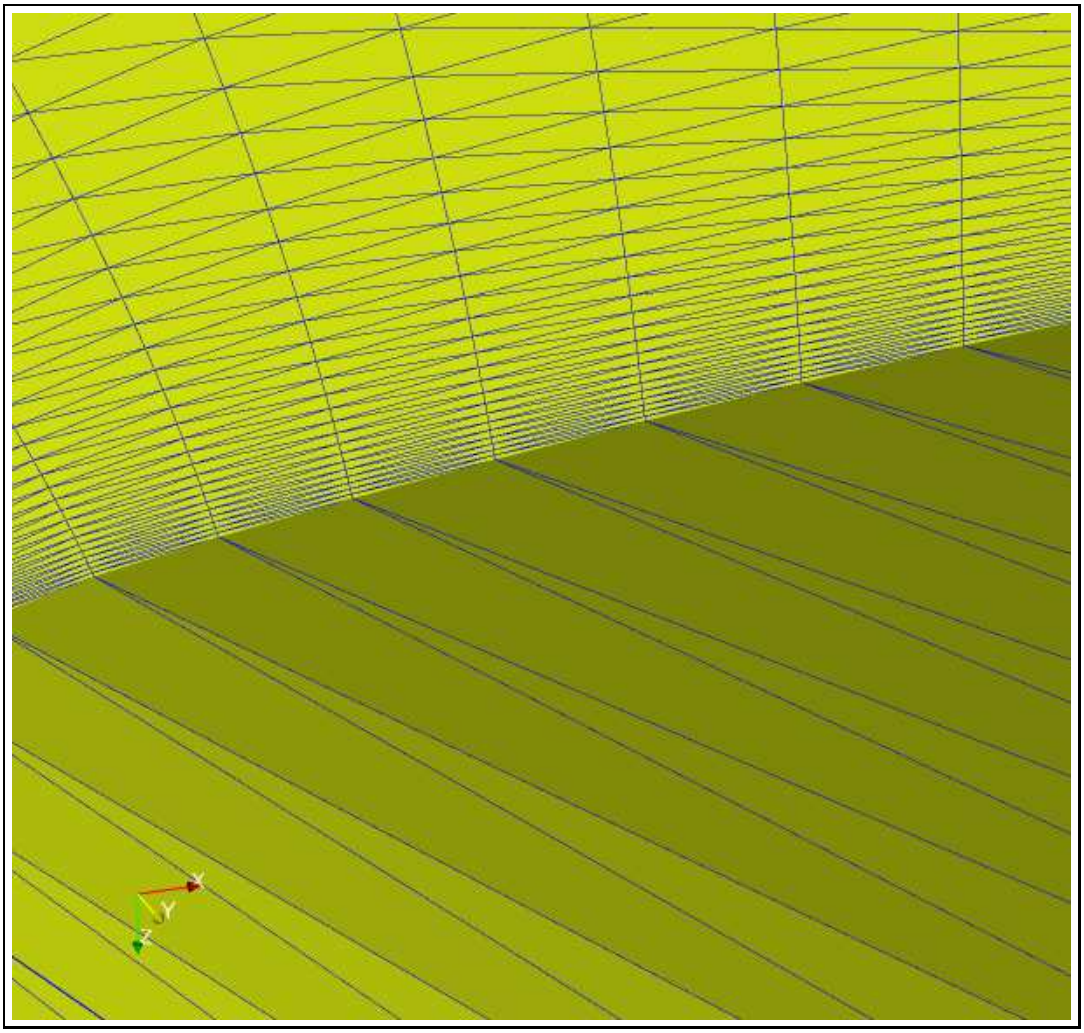


Figure 9: M6 wing : Zoom in the BL of the initial finest mesh

version for the first coarsened mesh than in figure 9. It is seen that the layered structure of the boundary layer is preserved by the coarsening algorithm.

Although table 2 gives a good indication on the process of removing the nodes, it is only a global measure. In order to have a better estimate on the efficiency of our semi-coarsening algorithm, figure 11 shows the histogram of the fourth generated coarse meshes together with the histogram of the initial finest mesh. It is seen that as coarsening progress, the first

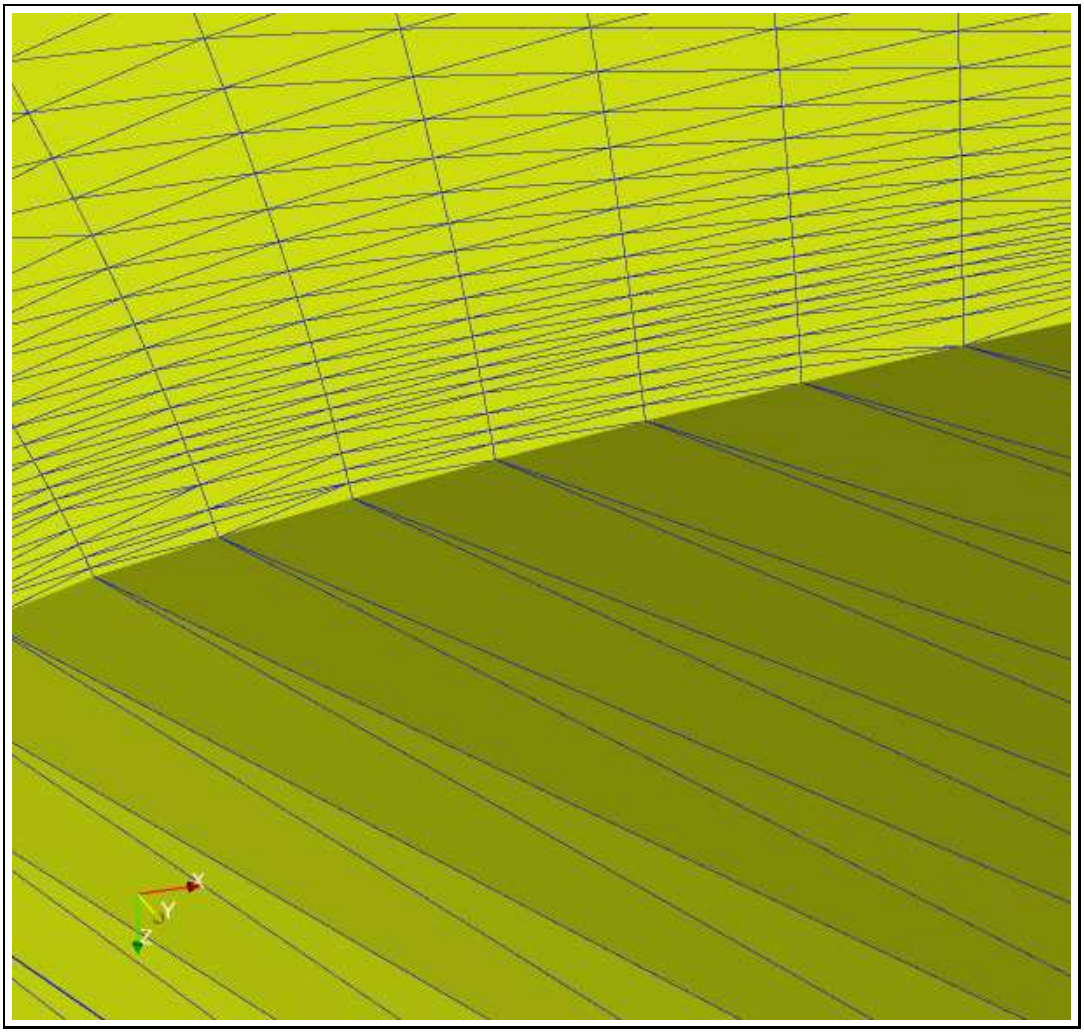


Figure 10: M6 wing : Zoom in the BL of the first coarsened mesh

classes containing highly stretched elements progressively disappear and that the meshes becomes more and more isotropic : The fourth coarsened mesh has less than 5% of its nodes with an aspect ratio larger than 39.

	Number of nodes	coarsening ratio
initial mesh	67866	
first coarsened mesh	30961	2.19
second coarsened mesh	15123	1.99
third coarsened mesh	8507	1.77
fourth coarsened mesh	5213	1.63

Table 2: Initial and coarsened meshes, number of nodes and coarsening ratio

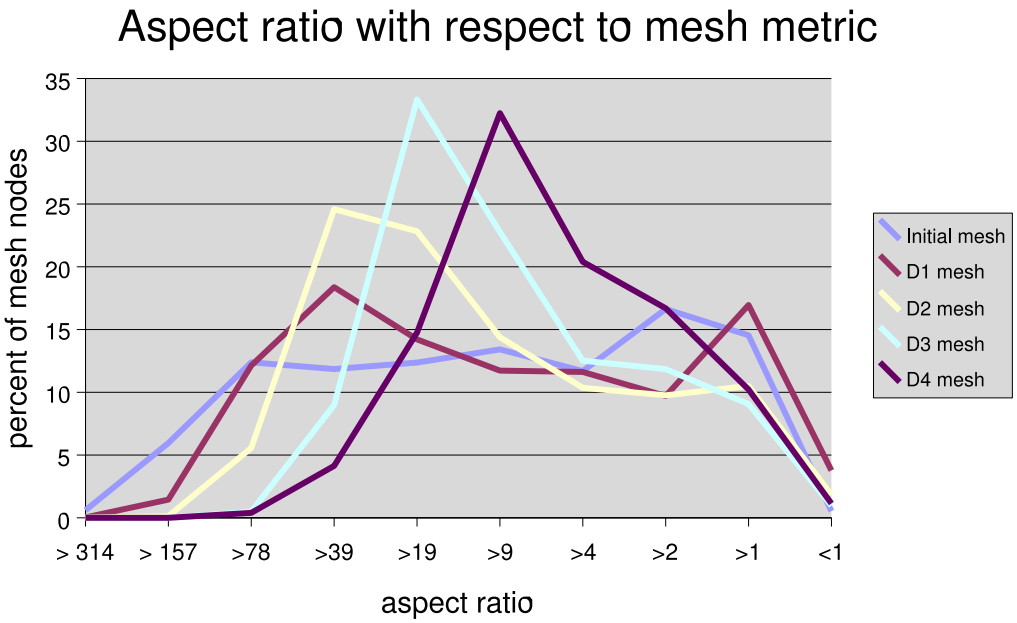


Figure 11: Initial M6 wing mesh

5.3 Falcon test case

We now move to an even more anisotropic mesh than the previous one. Figure 12 shows the geometry of a Falcon airplane used for this test case. The mesh of this test case is used for Navier-Stokes steady computations using Low Reynolds turbulence modeling. The mesh contains more than 2 millions nodes and is extremely stretched in the directions parallel to the geometry and the distance of the first point in the boundary layer normal to the surface can be as small as $10^{-9}m$. In these regions, the aspect ratio (ratio of the largest

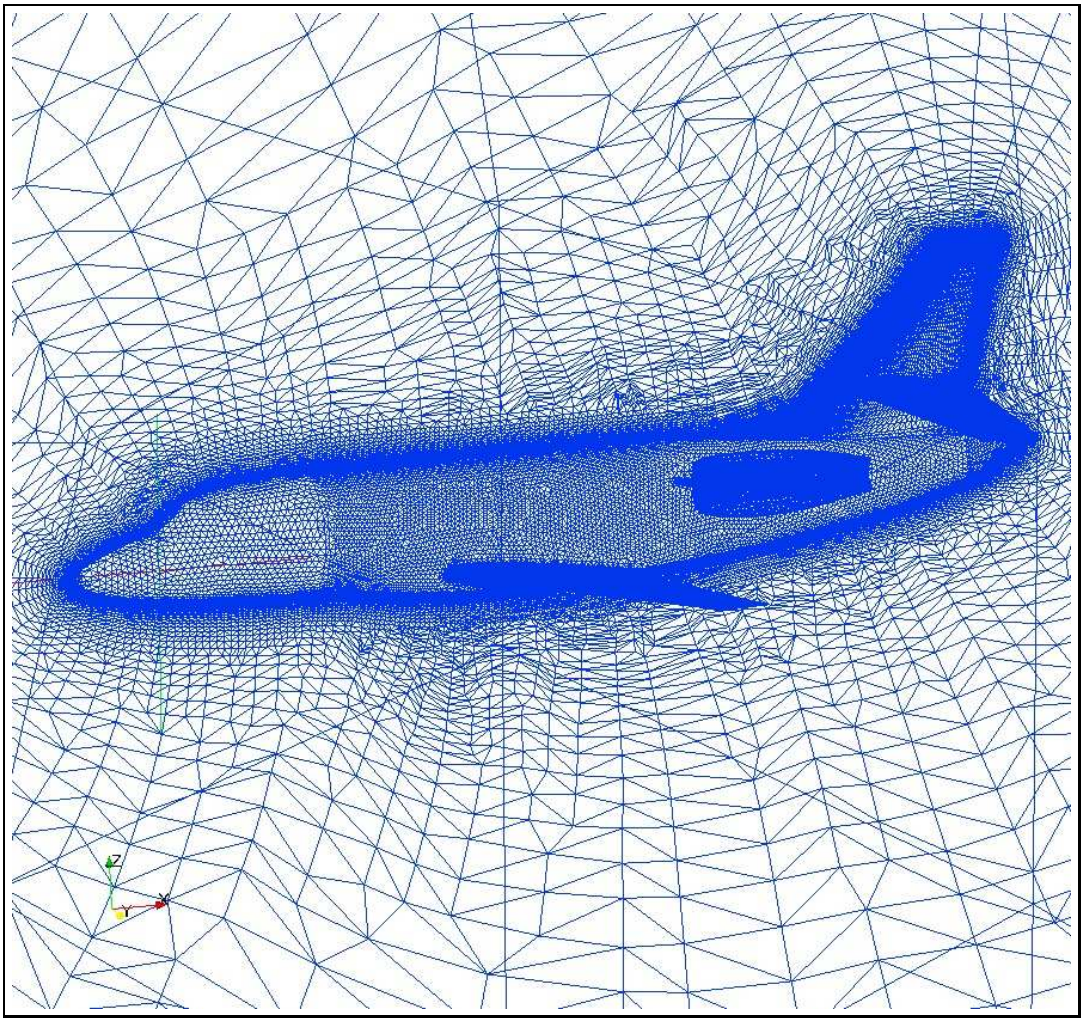


Figure 12: finest mesh

edge over the smallest) of the elements is extremely large. The maximum aspect ratio $ratio\_max$ for this mesh is extremely large and equal to 532713. To analyze this mesh, we have performed the same node classification than for the M6 mesh. Figure 13 displays the histogram corresponding to this node ordering. The y-coordinate corresponds to the number of nodes in each class divided by the total of nodes and 20 classes are present in this histogram. Table 3 gives a more accurate definition of these classes. It is seen that more

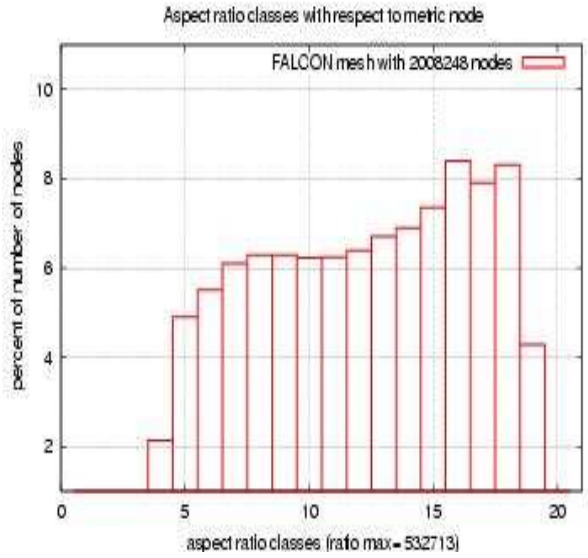


Figure 13: number of nodes (in % of the total number of nodes) with respect to aspect ratio classes

than 5% of the mesh nodes corresponds to an aspect ratio larger than 15 000. If we define isotropy as having an aspect ratio smaller than 5, this mesh has only 13 % of its node in the isotropic range (classes 18,19 20 see Table 3).

We have applied four times our mesh coarsening algorithm to this geometry. The figures [14] to [17] shows the different coarsened meshes successively obtained after four applications of the semi-coarsening algorithm.

	Number of nodes	aspect ratio interval
class 1	1	$532713 - 266356.5$
class 2	37	$266356.5 - 133178.25$
class 3	1192	$133178.25 - 66589.125$
class 4	42824	$66589.125 - 33294.562$
class 5	98511	$33294.562 - 16647.281$
class 6	110707	$16647.281 - 8323.640$
class 7	122646	$8323.640 - 4161.820$
class 8	124437	$4161.820 - 2080.910$
class 9	126260	$2080.910 - 1040.455$
class 10	125025	$1040.455 - 520.227$
class 11	125231	$520.227 - 260.113$
class 12	128334	$260.113 - 130.056$
class 13	134652	$130.056 - 65.028$
class 14	138429	$65.028 - 32.514$
class 15	147686	$32.514 - 16.257$
class 16	128581	$16.257 - 8.128$
class 17	158672	$8.128 - 4.064$
class 18	167013	$4.064 - 2.032$
class 19	85961	$2.032 - 1.016$
class 20	49	$1.016 - 0.508$

Table 3: table of the aspect ratio classes

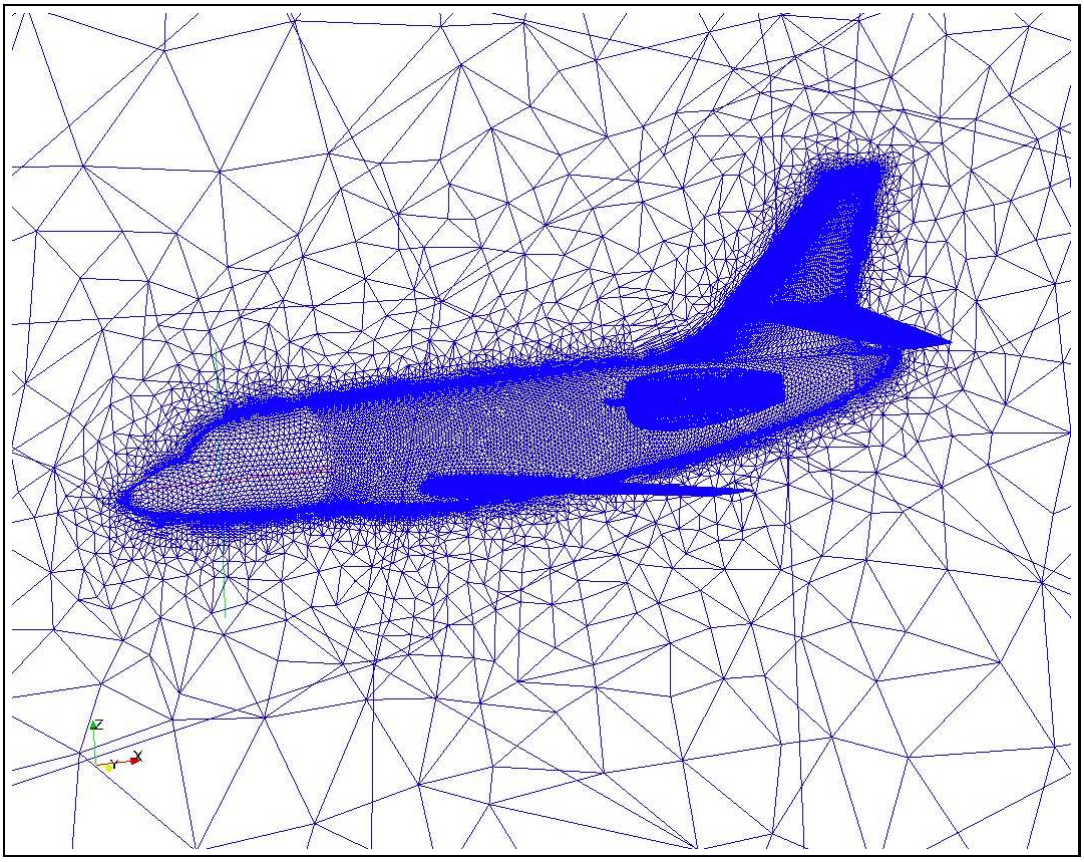


Figure 14: first coarsened mesh

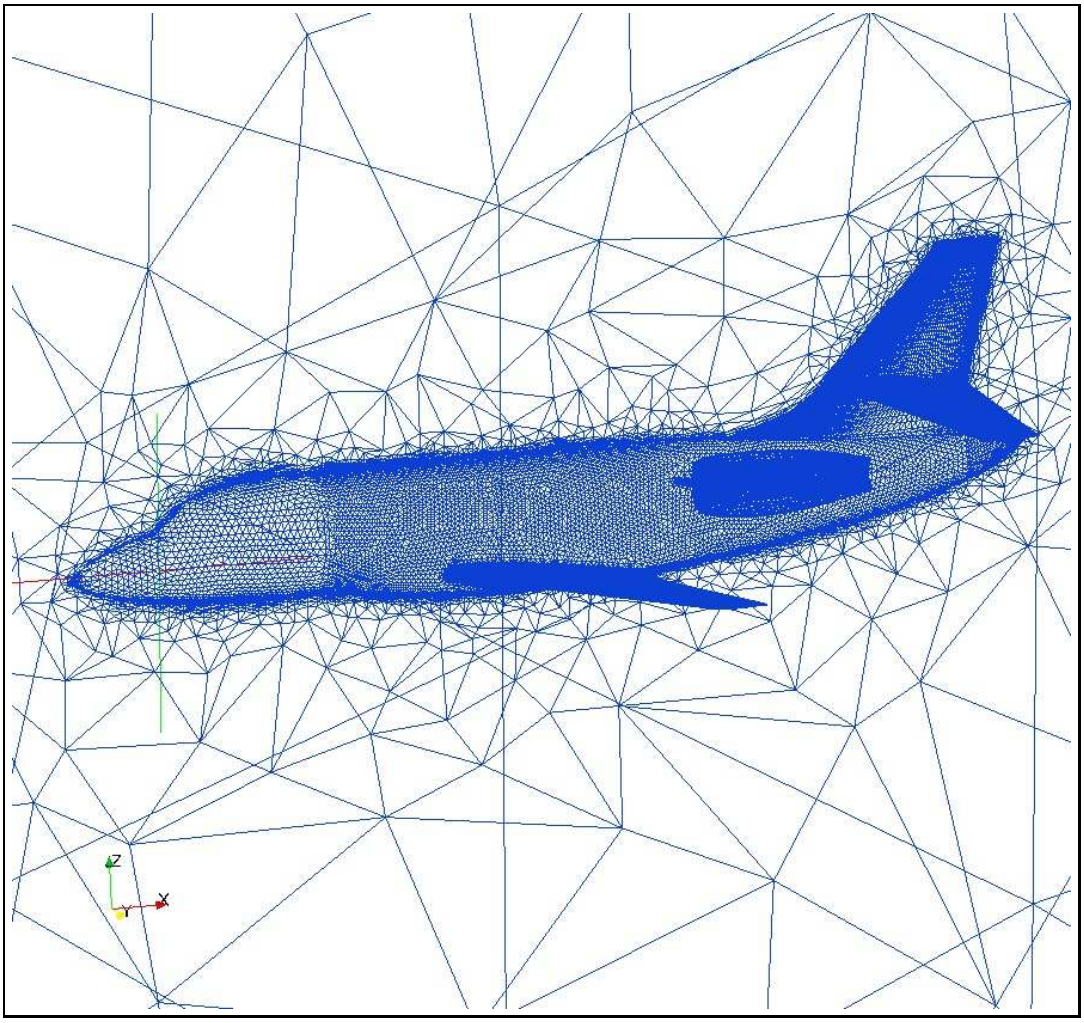


Figure 15: second coarsened mesh

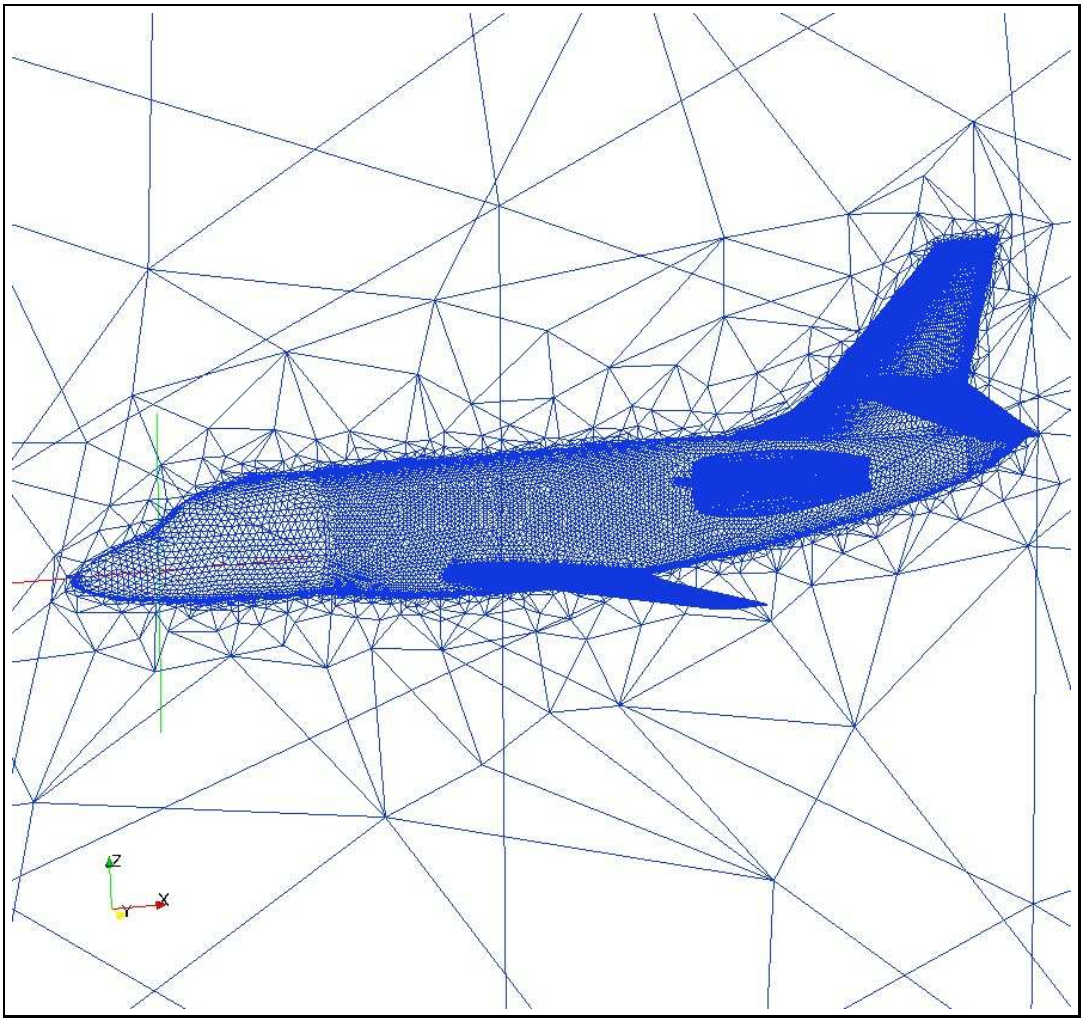


Figure 16: third coarsened mesh

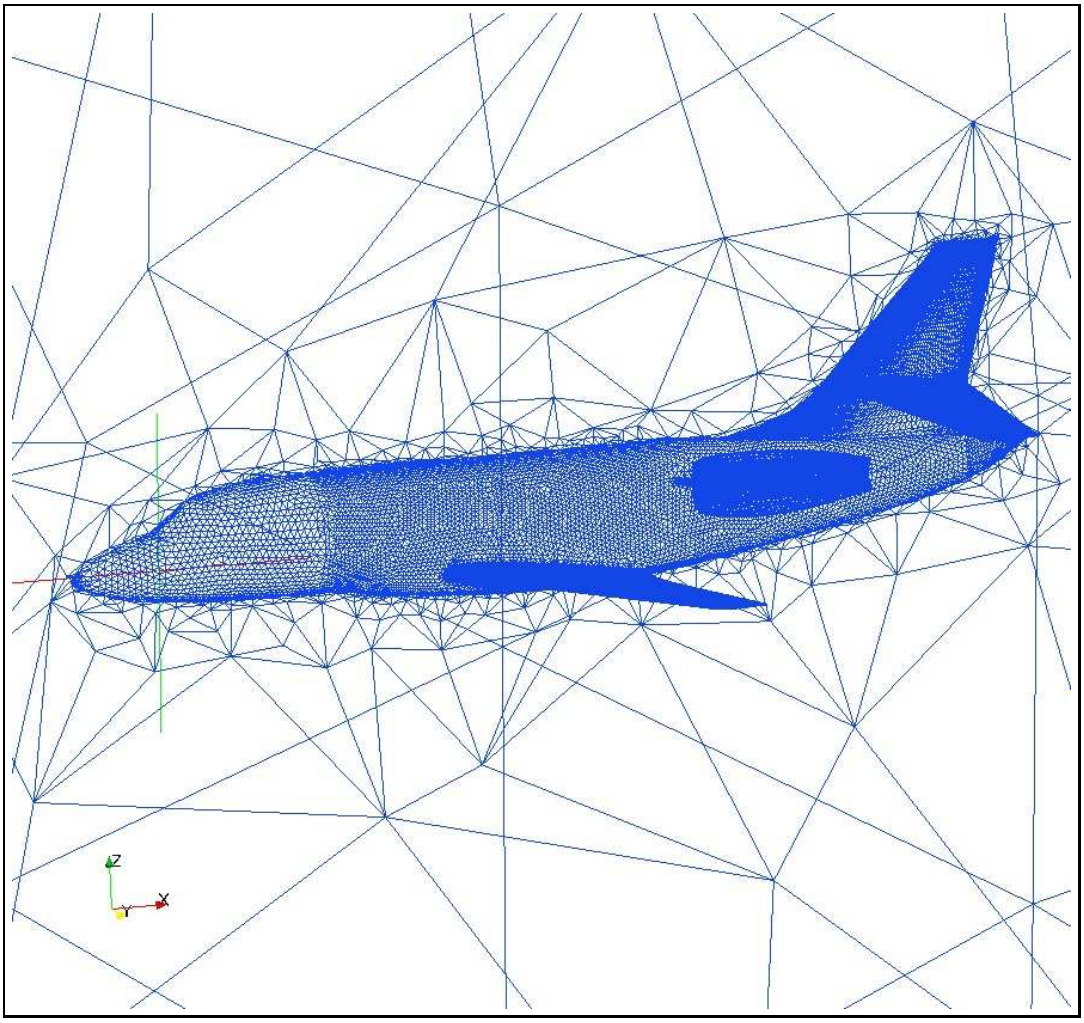


Figure 17: fourth coarsened mesh

	Number of nodes	coarsening ratio
initial mesh	2008248	
first coarsened mesh	779558	2.57
second coarsened mesh	338762	2.3
third coarsened mesh	218850	1.54
fourth coarsened mesh	139522	1.56

Table 4: table of initial and coarsened meshes, number of nodes and coarsening ratio

The first conclusion is that the mesh coarsener is indeed able to generate a sequence of meshes with decreasing number of nodes even with these highly stretched meshes. Table 5.3 gives another description of the process of mesh coarsening on the different levels. The coarsening ratio is approximately equal to 2 confirming the success of our semi-coarsening strategy. Examination of table 5.3 also shows that the coarsening factor initially high (2.45) gradually decreases and that the coarsening algorithm have difficulties to remove the nodes according to the prescribed metric field. As for the M6 mesh, the reason is very likely related to the geometrical quality of the mesh. As less and less nodes are present in the different meshes, it becomes more difficult to remove some of them since their removal would imply non-valid triangulations.

Figure 18 displays the histogram corresponding to the node ordering of the different meshes according to their aspect ratio. It is seen that as the directional coarsening proceeds, the high aspect ratio classes progressively disappear and that the meshes becomes more and more isotropic. This again confirm the success of our semi-coarsening algorithm. Note however that after four successive coarsening the final mesh remains highly anisotropic with a maximum aspect ratio of 82859.2 and only 30 % of the mesh in the isotropic range (i.e aspect ratio less than 6).

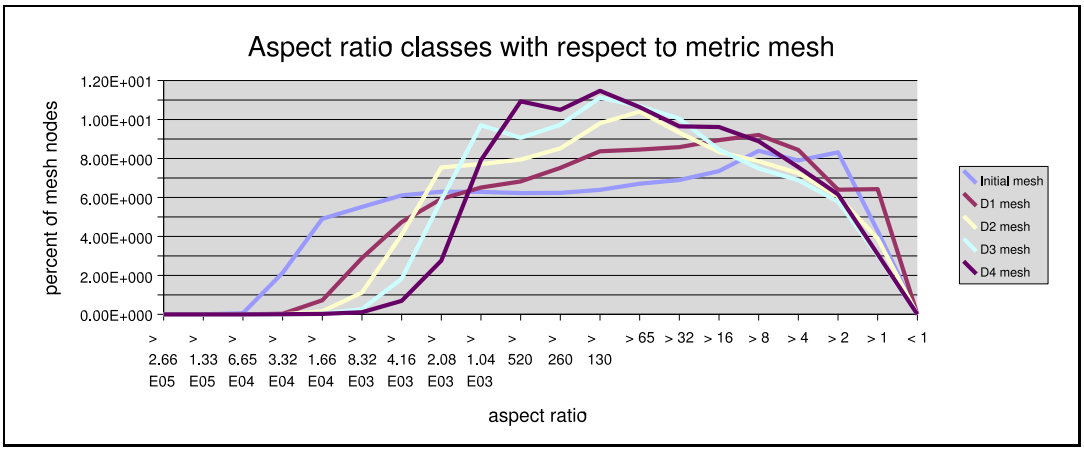


Figure 18: Number of vertices (in % of the total number of nodes) belonging to the corresponding Aspect ratio class

6 Conclusion

The present work has demonstrated the capability of our semi-coarsening algorithm to handle extremely stretched meshes used for Navier-Stokes computations with Low Reynolds turbulence modeling. This algorithm is able to generate a sequence of meshes with decreasing number of nodes in the direction perpendicular to the geometry. This sequence of meshes are thus suitable for Multigrid acceleration.

In particular in this work we have :

1. study the interpolation procedures and smoothing of the metric fields. Our work has relied on some recent results on metric interpolations [10, 11] as well as on some more simpler ideas based on averaging of the mesh edges length. In addition, we have proposed a way to handle the connection between isotropic regions in the far field and the anisotropic ones in the boundary layers.

2. restructured the MTC mesh generator and proposed new definition of the quality functions used in this software that are more adapted to our purpose. Moreover, this mesh generation tool is now currently able to handle large meshes since we have optimized the size of the work arrays in order to save memory.

3. performed several relevant tests for the semi-coarsening strategy. These experiments are extremely encouraging and show that Multigrid methods can be used for the meshes employed for Navier-Stokes computations with turbulence modeling in an industrial context.

Several remarks arise also during this work.

1. Due to the extremely large aspect ratio used in Navier-Stokes computations and since semi-coarsening uses a coarsening factor of 2, it is seen that 20 different coarsened meshes will have to be generated before reaching isotropy. With the need to handle these meshes, the cost of a MG algorithm will be high and problems of mesh storage can arise. It therefore could be of interest to study more aggressive coarsening strategies in order to reduce more rapidly the size of the coarsened meshes as well as to study the influence of larger coarsening factors on multigrid convergence.

2. The coarsening strategy have certainly to be optimized in order to maintain a constant coarsening factor between meshes. This can be realized by different methods : change of the coarsening factor from levels to levels, use of selected meshes in the set of generated meshes and so on. Studies have to be performed in order to measure the influence of these adaptations on multigrid convergence and the CPU cost of the computations.

A Computation and interpolation of the metric field

The computation of the metric fields described in this paper, has been implemented into a C++ software denoted AMeCO (Anisotropic Mesh Coarsener). AMeCo is an object oriented software, that computes efficiently the metric fields over several anisotropic unstructured 3D meshes. This software is designed to be automatic and user-friendly.
The generation of the coarse levels is done by recursively calling the AMeCo and the mesh generation software as described in figure (19). However, the input/output data of both softwares are different, so a conversion interface between the two tools is required. For this reason, we have developed a conversion tool that converts input/output formats from each software format to other. Furthermore, the INRIA and Dassault data formats are not the same, so we have added in the converter the option that allows to switch from different formats. From the user point of view, all these treatment are managed and encapsulated on a meta-script and then the coarsening process is also automatic. Let us now describing the

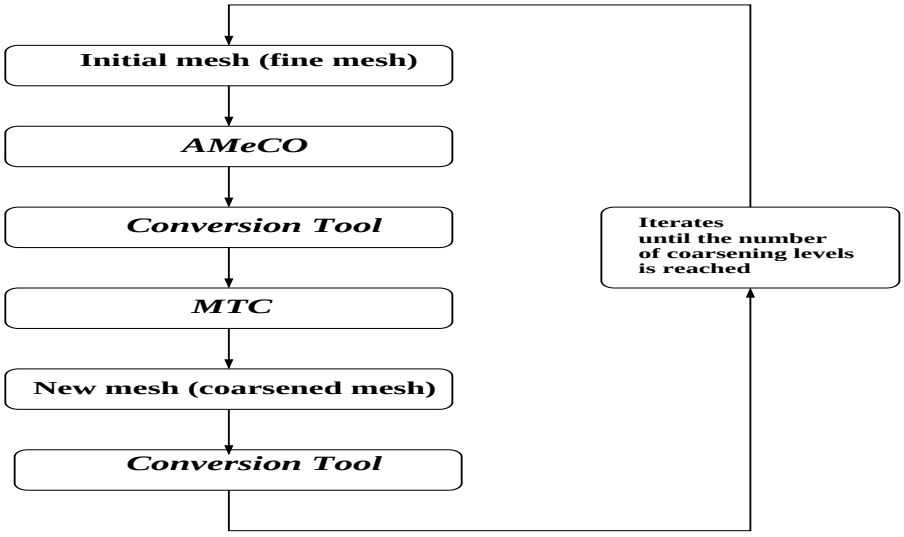


Figure 19: Coarsening process

three formats as used in the coarsening process:

A.1 MTC format

A.1.1 MTC meta-format

MTC software has as input a meta-file denoted file.mtc, where the suffix file corresponds to mesh name or user notation. This meta-file describes the file mesh name (file.t) and the metric file name. For example, if the user has to coarsen a M6 mesh, he will use for that a

meta-file denoted M6.mtc that reads:

```
{ M:          M6.t }
{ Metrique:   M6_metrique }
```

where M: and Metrique: are keywords identifying for MTC the location of the files describing the mesh and associated metric field, M6.t is the mesh file in MTC format, and M6_metrique is the file that contains metrics. These files are described below:

A.1.2 MTC mesh format

The MTC mesh file (file.t) is an ascii data file and is written as follows:

nb_nodes	dim_space	$nb\_elements$	$dim\_element$
x_1^1	.	$x_{dim\_space}^1$	
.	.	.	
.	.	.	
x_1^i	.	$x_{dim\_space}^i$	
.	.	.	
.	.	.	
$x_1^{nb\_nodes}$	.	$x_{dim\_space}^{nb\_nodes}$	
e_1^1	.	.	$e_{dim\_element}^1$
.	.	.	.
.	.	.	.
e_1^i	.	.	$e_{dim\_element}^i$
.	.	.	.
.	.	.	.
$e_1^{nb\_elements}$	.	.	$e_{dim\_element}^{nb\_elements}$

where nb_nodes is the number of nodes, dim_space the space dimension, $nb\_elements$ is the number of elements (tetrahedrons in 3D and triangles in 2D), $dim\_element$ is the number of nodes by element, x_j^i is the jth coordinate of ith node, and e_j^i is the jth node of ith element.

Remark: The elements in MTC format contains also the boundary faces, the difference between an internal element and a boundary element that the fourth node number in a boundary face has the index 0.

A.1.3 MTC metric format

The metric data file is an ascii data files contains metrics of each node on the mesh:

$$
\begin{array}{ccccc}
nb\_nodes & & & & \\
m_{11}^{1} & . & m_{ij}^{1} & m^{1}_{dim\_spacedim\_space} \\
. & . & . & . \\
. & . & . & . \\
m_{11}^{i} & . & m_{ij}^{k} & m^{i}_{dim\_spacedim\_space} \\
. & . & . & . \\
. & . & . & . \\
m_{11}^{nb\_nodes} & . & m_{ij}^{nb\_nodes} & m^{nb\_nodes}_{dim\_spacedim\_space}
\end{array}
$$

where nb_nodes is the number of nodes, dim_space the space dimension, m_{ij}^{k} is the ijth component of the metric m on the k.

A.2 AMeCo format

The AMeCO mesh file (file.dimeco) is a binary data file and is written as follows:

nb_nodes	$nb\_elements$	nb_faces	
x_1^1	.	$x^1_{dim\_space}$	
.	.	.	
.	.	.	
x_1^i	.	$x^i_{dim\_space}$	
.	.	.	
.	.	.	
$x_1^{nb\_nodes}$	.	$x^{nb\_nodes}_{dim\_space}$	
e_1^1	.	.	$e^1_{dim\_element}$
.	.	.	.
.	.	.	.
e_1^i	.	.	$e^i_{dim\_element}$
.	.	.	.
.	.	.	.
$e_1^{nb\_elements}$	.	.	$e^{nb\_elements}_{dim\_element}$
f_1^1	.	.	$f^1_{dim\_face}$
.	.	.	.
.	.	.	.
f_1^i	.	.	$f^i_{dim\_face}$
.	.	.	.
.	.	.	.
$f_1^{nb\_faces}$	.	.	$f^{nb\_faces}_{dim\_face}$
$logic\_fac^1$			
.			
.			
$logic\_fac^i$			
.			
.			
$logic\_fac^{nb\_faces}$			

where nb_nodes is the number of nodes, $nb\_elements$ is the number of elements (tetrahedrons in 3D and triangles in 2D), nb_faces is the number of boundary faces, x_j^i is the jth coordinate of ith node, e_j^i is the jth node of ith element, f_j^i is the jth node of the ith face, and $logic\_fac^i$ is the logical boundary of the ith face .

A.3 Dassault format

The Dassault mesh file (file.dassault) is an ascii data file and is written as follows:

nb_nodes	$nb\_elements$	nb_faces	
x_1^1	.	$x^1_{dim\_space}$	
.	.	.	
.	.	.	
x_1^i	.	$x^i_{dim\_space}$	
.	.	.	
.	.	.	
$x_1^{nb\_nodes}$	.	$x^{nb\_nodes}_{dim\_space}$	
e_1^1	.	.	$e^1_{dim\_element}$
.	.	.	.
.	.	.	.
e_1^i	.	.	$e^i_{dim\_element}$
.	.	.	.
.	.	.	.
$e_1^{nb\_elements}$	.	.	$e^{nb\_elements}_{dim\_element}$
f_1^1	.	$f^1_{dim\_face}$	$logic\_fac^1$
.	.	.	.
.	.	.	.
f_1^i	.	$f^i_{dim\_face}$	$logic\_fac^i$
.	.	.	.
.	.	.	.
$f_1^{nb\_faces}$	.	$f^{nb\_faces}_{dim\_face}$	$logic\_fac^{nb\_faces}$

where nb_nodes is the number of nodes, $nb\_elements$ is the number of elements (tetrahedrons in 3D and triangles in 2D), nb_faces is the number of boundary faces, x_j^i is the jth coordinate of ith node, e_j^i is the jth node of ith element, f_j^i is the jth node of the ith face, and $logic\_fac^i$ is the logical boundary of the ith face .

B Differential manifolds and the Riemannian geometry

The purpose of this appendix is to familiarize the reader with some essential concepts on the differential manifolds and the Riemannian geometry in an intuitive way but is not in no case a complete course nor rigorous on the subject. For that, I return the reader to the work of Manfredo Perdigao do Carmo, Riemannian Geometry [20]. Few concepts bases which will be described in this appendix are essential to introduce a Riemannian framework for the tensors.

B.1 Definitions

Intuitively, a manifold is a " curved " space of R^N. As it is difficult to represent a curved space in addition to 3 dimensions, our examples will be limited to manifolds of R^3. For example, a sphere or a torus are manifolds of R^3 or in general, any surface is a manifold of R^3. The space of the tensors Sym_n^+ which we will describe thereafter is a manifold of R^6.

Let us take the example of the sphere S_2. The sphere is a manifold of R^3. A torus, a quadric are manifolds of R^3. If one observes a sphere, one can approximate the curvature of its surface by a tangent plan. It is exactly what us let us make when a geographical chart is drawn: the curve of the ground is approached by a plan.

Definition 1: Differentiable manifold
Informally, a differentiable manifold is a type of manifold (which is in turn a kind of topological space) that is similar enough to Euclidean space to allow one to do calculus. It is important to note that differentiable can mean slightly different things in different contexts, such as continuously differentiable, k times differentiable, or infinitely differentiable (also known as smooth).

Let us take again our example of the sphere. For each point of this manifold, one can be defines a tangent plan. This tangent plan contains an infinity of tangent vectors to the manifold. With the illustration of figure 20, one will understand that one can add two vectors of a tangent plan and to remain in this plan.

In the same way, one can multiply a vector by one scalar and to remain in the plan. On the other hand, it is impossible to add two vectors of two different tangent plans because one would leave one or the other plan.
Finally, one can show that a tangent plan has a structure of vector space, that one will be called tangent space T_xM to the manifold M at a point x.

Definition 2: Riemannian manifold
a Riemannian manifold (M,g) (with Riemannian metric g) is a real differentiable manifold M in which each tangent space is equipped with an inner product g in a manner which varies smoothly from point to point. This allows one to define various notions such as angles, lengths of curves, areas (or volumes), curvature, gradients of functions and divergence of

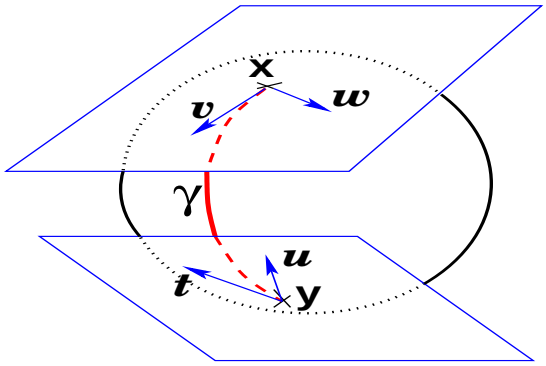


Figure 20: The tangent plants at sphere points x and y are differents: the vectors v and w of T_xM can not be compared with t and u of T_yM. Then the scalar product is defined on each tangent plan

vector fields. In other words, a Riemannian manifold is a differentiable manifold in which the tangent space at each point is a finite-dimensional Hilbert space.
One speaks then of Riemannian geometry when one works on this type of manifolds.

Definition 3: Riemannian manifold as metric space
If $\gamma : [a, b] \rightarrow M$ is a continuously differentiable curve in the Riemannian manifold M (example of curve $\gamma(t)$ on the sphere S_2 (figure 20).), then we define its length $L(\gamma)$ by

$$L(\gamma) = \int_a^b ||\gamma'(t)||dt$$

With this definition of length, every connected Riemannian manifold M becomes a metric space (and even a length metric space) in a natural fashion: the distance $d(x, y)$ between the points x and y of M is defined as

$$d(x, y) = inf\{L(\gamma) : \gamma \text{ is a continuously differentiable curve joining } x \text{ and } y\}$$

Even though Riemannian manifolds are usually "curved", there is still a notion of "straight line" on them: the geodesics. These are curves which locally join their points along shortest

paths. Assuming the manifold is compact, any two points x and y can be connected with a geodesic whose length is $d(x, y)$.

B.2 Exponential chart

Let x be a point of a manifold M and $\vec{xy}$ a vector of tangent space T_xM at that point. We know (think to second order differential equations theory) that there exists one and only one geodesic starting at x and with $\vec{xy}$ as a tangent vector. This allows to develop M on tangent space T_xM (imagine rolling a sphere along its tangent space at given point) (fig. 21). The geodesics going through the reference point are transformed in a straight lines and the distance along these geodesics are conserved.
The function which corresponds to each vector $\vec{xy}$ of tangent space, the point y of the manifold reached after a unit time by the geodesic staring at x and with this tangent vector is called the *exponential map:* $y = exp_x()$. This function carries out a local diffeomorphism of tangent space T_xM on M.

The inverse of the exponential map, the logarithmic map: $\vec{xy} = log_x(y)$ corresponds to

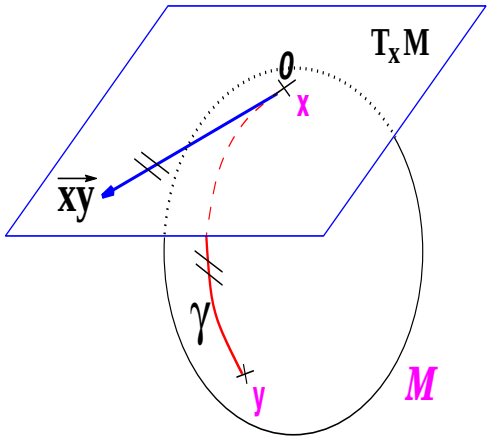


Figure 21: The geodesics starting at x are stright lines in the exponenetial map and the distance along them are conserved

all bipoint (x, y) of M, the tangent vector $\vec{xy}$ such as the geodesic joining x to y has as a tangent vector $\vec{xy}$ at x.

In the general case, the logarithmic map is defined everywhere on the manifold except on cut-locus C_x: it is the closure of the set of points where several geodesics starting at x meet. It is the case of the antipodean points on the sphere S_2 where there is an infinity of geodesics connecting them (figure 22). Consequently, the cut-locus C_x of any point x is its antipodean point and the tangent cut-locus is the circle of radius π (figure 22). The logarithmic chart is defined on $M - C_x M$.
One supposes in the sequel that M does not take cut-locus and is geodesically complete, i.e. for any couple (x, y) of M there exists one and only one geodesic connecting them.

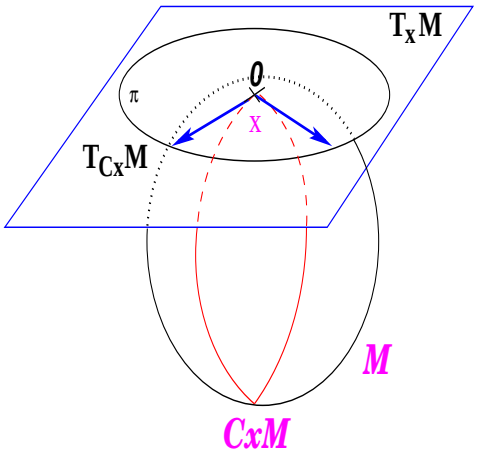


Figure 22: The place of cut of the point x of S_2. An infinity of geodesics joining x and its antipodean point (two are represented in red). If one unfolds all geodesics on tangent space $T_x M$, one finds the circle of radius π. The logarothmic chart is defined on $M - C_x M$.

B.3 Re-interpretation of the basic operations

We saw in the previous sections that the exponential chart allows to make correspond to any tangent vector a point of M and that reciprocally, the logarithmic chart makes the correspondence between two point (x, y) of M and a tangent vector. The goal is to use the structure of vector space of $T_x M$ to carry out our calculations like we know to do it. With through the two functions which we have just seen, it is possible for us to generalize the operations of addition and subtraction to Riemannian manifolds. The difficulty is to remain

on the manifold for any operation. The following table shows some basic operations of the vector spaces and their correspondences in Riemannian geometry:

Vector space	Riemannian manifold
$\vec{xy} = y - x$	$\vec{xy} = log_x(y)$
$y = x + \vec{xy}$	$y = exp_x(\vec{xy})$
$dist(x,y) = \|\|y - x\|\|$	$dist(x,y) = \|\|\vec{xy}\|\|_x$

Table: Reinterpretation of addition and soustraction on a Riemannian manifold.

These basic operations will enable us to generalize with the Riemanian manifolds many algorithms reserved until there for the vector spaces. Implementation of logarithmic and exponential charts is the key of this generalization since all geometrical operations can be expressed starting from these operations.

This section on the reinterpretation of the basic operations puts a term at the recalls on the differential manifolds and the Riemannian geometry. In the following section, we will see the construction of a Riemannian metric, the geodesics and exponential and logarithmic charts for the space of the tensors which we will use to define a rigorous mathematical framework to handle this type of matrices.

B.4 Working on the tensor space

Let us now focus on the space Sym_n^+ of positive definite symmetric matrices (tensors). Its worth to notice that the space Sym_n^+, although a sub-set of a vector space, is not a vector space, .e.g., the negation of a positive-definite matrix is not positive-definite.
The goal is to find a Riemannian metric with interesting enough properties. It turns out that it is possible to require an invariance by the full linear group (Section B.4.1). This leads to a very regular manifold structure where tensors with null and infinite eigenvalues are both at an infinite distance of any positive definite symmetric matrix: the cone of positive definite symmetric matrices is replaced by a space which has an infinite development in each of its $n(n+1)/2$ directions. Moreover, there is one and only one geodesic joining any two tensors. Thus, the structure we obtain is very close to a vector space, except that the space is curved.

B.4.1 An affine invariant distance

Let us consider the following action of the linear group $GL_n = \{M \in M_{n*n}/det(M) \neq 0\}$ on the tensor space Sym_n^+ :

$$A \diamond \Sigma = A\Sigma A^T \ \forall A \in GL_n \text{ and } \Sigma \in Sym_n^+ \tag{20}$$

This action is naturally extended to tangent vectors: if $\Gamma(t) = \Sigma + tW + \mathcal{O}(t^2)$ is a curve passing at Σ with tangent vector W, then the curve $A \diamond \Gamma(t) = A\Sigma A^T + \mathcal{O}(t^2)$ passes trough

$A \diamond \Sigma$ with tangent vector $A \diamond W$.
Following [18], any invariant distance on Sym_n^+ verifies $dist(A \diamond \Sigma_1, A \diamond \Sigma_2) = dist(\Sigma_1, \Sigma_2)$. Choosing $A = \Sigma_1^{-1/2}$, we can reduce this to a pseudo-norm, or distance to identity:

$$dist(\Sigma_1, \Sigma_2) = dist(Id, \Sigma_1^{-/2} \Sigma_2 \Sigma_1^{-/2}) = N(\Sigma_1^{-/2} \Sigma_2 \Sigma_1^{-/2}). \tag{21}$$

Moreover, as the invariance has to hold for any transformation, N should be invariant under the action of the isotropy group $H(Id) = \mathcal{O}_n = U \in GL_n / UU^T = Id$:

$$\forall U \in \mathcal{O}_n, N(U \Sigma U^T) = N(\Sigma). \tag{22}$$

Using the spectral decomposition $\Sigma = UD^2U^T$, it is easy to see that $N(\Sigma)$ has to be symmetric function of the eigenvalues. Moreover, the symmetry of the distance $dist(\Sigma, Id) = dist(\Sigma, Id)$ imposes that $N(\Sigma) = N(\Sigma^{-1})$. Thus, a good candidate is the sum of the squared logarithms of the eigenvalues:

$$N(\Sigma^2) = \sum_{i=1,n} (log(\lambda_i))^2. \tag{23}$$

This "norm" verify by construction the symmetry and positiveness. $N(\Sigma) = 0$ implies that $\lambda_i = 1$ (and conversly), so the separation axiom is verified. However, we do no know any simple proof of the triangle inequality, which should read $N(\Sigma_1 + \Sigma_2) \leq N(\Sigma_1) + N(\Sigma_2)$, even if we can verify it experimentally.

Remark: The tangent space of Sym_n^+ is the space of symmetric matrices $Sym(n)$. Indeed, since the group action $\phi_A : \Sigma \rightarrow A \Sigma A^T$ is linear, its derivative map, denoted $d\phi_A$, is given by $d\phi_A = AXA^T$. If $X \in Sym(n)$, it is easy to see that $d\phi_A$ is again a symmetric matrix. Thus the tangent space at any point $\Sigma \in Sym_n^+$ is also equivalent to $Sym(n)$.

B.4.2 An invariant Riemannian metric

Another way to determine the invariant distance is through the Riemannian metric. Let us take the most simple scalar product on the tangent space at the identity matrix: if W_1 and W_2 are tangent vectors (i.e. symmetric matrices, not necessarily definite nor positive), we define the scalar product to be the standard matrix scalar product $\langle W1 | W2 \rangle = Tr(W_1^T W2)$. This scalar product if obviously invariant by the isotropy group $\mathcal{O}_n$. Now, if W_1 and W_2 are two tangent vectors at Σ, we require their scalar product to be invariant by the action of any transformation: $\langle W1 | W2 \rangle = \langle A \diamond W1 | A \diamond W2 \rangle_{A \diamond \Sigma}$. This should be true in particular for $A = \Sigma^{-1/2}$, which allows us to define the scalar product at any Σ from the scalar product at the identity:

$$\langle W1 | W2 \rangle_\Sigma = \langle \Sigma^{-1/2} W_1 \Sigma^{-1/2} | \Sigma^{-1/2} W_2 \Sigma^{-1/2} \rangle_{Id} = Tr(\Sigma^{-1/2} W_1 \Sigma^{-1} W_2 \Sigma^{-1/2}) \tag{24}$$

Geodesics on tensor manifold going through Σ with tangent vector W should have the following form (see Pennec for more details):

$$\Gamma_{(\Sigma, W)}(t) = \Sigma^{1/2} exp(t \Sigma^{-1/2} W \Sigma^{-1/2}) \Sigma^{1/2}$$

B.4.3 Exponential and logarithm map

As a general property of Riemannian manifolds, geodesics realize a local diffeomorphism from the tangent space at a given point of the manifold to the manifold: $\Gamma_{(\Sigma,W)}(1) = exp_\Sigma(W)$ associates to each tangent vector $W \in T_\Sigma Sym_n^+$ a point of the manifold. This mapping is called the exponential map, because it corresponds to the usual exponential in some matrix groups. This is exactly our case for the exponential map around the identity:

However, the Riemannian exponential map associated to our invariant metric has a more complex expression at other tensors:

$$exp_{Id}(UDU^T) = exp(UDU^T) = Uexp(D)U^T. \tag{25}$$

However, the Riemannian exponential map associated to our invariant metric has a more complex expression at other tensors:

$$exp_\Sigma(W) = \Sigma^{1/2} exp(\Sigma^{-1/2} W \Sigma^{-1/2}) \Sigma^{1/2}. \tag{26}$$

Thus, exp_Σ gives us a collection of one-to-one and complete maps of the manifold, centered at any point Σ. As explained in Section 2.1, these charts can be viewed as the development of the manifold onto the tangent space along the geodesics. Moreover, as the manifold has a non-positive curvature [18], there is no cut-locus and the statistical properties detailed in [18] hold in their most general form. For instance, we have the existence and uniqueness of the mean of any distribution with a compact support.

References

[1] GUILLARD H., *Node-Nested Multi-Grid with Delaunay Coarsening* INRIA Research Report N° 1898, 1993.

[2] JANKA A. AND GUILLARD H., *developpement et tests de méthodes multigrilles pour l'accélération de la convergence des méthodes de résolution de Navier-Stokes sur maillages non-structurés* Final Report INRIA Sophia-Antipolis SMASH Project, 2003.

[3] COUPEZ T., *Génération de maillage et adaptation de maillage par optimisation locale* Revue européene des éléments finis 9 (2002), no. 4, 403-423.

[4] CARTE G., COUPEZ T, GUILLARD H AND LANTERI S., *Coarsening techniques in multigrid applications on unstructured meshes* european congress on computational methods in applied sciences and engineering, ECCOMAS 2000, Barcelona, 2000.

[5] H. BOROUCHAKI, P. GEORGE, F. HECHT, P. LAUG, E. SALTEL, *Delaunay mesh generation governed by metric specifications. Part 1 : Algorithms*, Finite Elements in Analysis and Design, 25, 1997, p. 61-83.

[6] H. BOROUCHAKI, P. GEORGE, B. MOHAMMADI *Delaunay mesh generation governed by metric specifications. Part 2 : Application examples*, Finite Elements in Analysis and Design, 25, 1997, p. 85-109.

[7] P. GEORGE, H. BOROUCHAKI, *Premières expériences de maillage automatique par une méthode de Delaunay anisotrope en trois dimensions*, Technical report 0272, INRIA, 2002, http://www.inria.fr/rrrt/rt-0272.html.

[8] H. BOROUCHAKI, F. HECHT AND P. FREY *Mesh Gradation Control*, International Journal for Numerical Methods in Engineering, Wiley, Vol 43, pp.1143-1165, 1999

[9] GEORGE P.L AND BOROUCHAKI H., *Triangulation de Delaunay et maillage, applications auc éléments finis* Hermès, Paris

[10] ARSIGNY, VINCENT - FILLARD, PIERRE - PENNEC, XAVIER - AYACHE, NICHOLAS, *Fast and Simple Computations on Tensors with Log-Euclidean Metrics* Rapport de recherche de l'INRIA - Sophia Antipolis, RR-5584.

[11] FLETCHER, P.T. AND JOSHI, S., *Principal Geodesic Analysis on Symmetric Spaces: Statistics of Diffusion Tensors* A trouver.

[12] FRANCESCATTO J, AND DERVIEUX A., *A semi-coarsening strategy for unstructured multigrid method based on Agglomeration* Int. J. Num. Mesh. Fluids 26, 927-957.

[13] LALLEMAND M.H., STÈVE H., AND DERVIEUX A., *Unstructured multigriding by Volume Agglomeration: Current Status* Computers and Fluids 21, 397-433.

[14] Koobus B., Lallemand M.H., and Dervieux A., *Unstructured Volume Agglomeration MG : Solution of Poisson Equation* Int. J. Num. Mesh. Fluids 18, 27-42.

[15] Mesri Y., and Guillard H., *Méthodes de déraffinement anisotrope pour la construction d'une hiérarchie de maillages d'une géométrie 3D* Dassault Report.

[16] Fréchet M. *Les éléments aléatoires de nature quelconque dans un espace distancié* Inst. H. Poincaré, (10):215-310,1948

[17] Karcher H. *Riemannian center of mass and mollifier smoothing* Communications on Pure and Applied Mathematics, 30(5):509-541, 1977

[18] Pennec X. *Probabilities and statistics on Riemannian manifolds: basic tools for geometric measurements.* IEEE workshop on non-linear signal and image processing, 1999.

[19] Borouchaki H., Hecht F, and Frey P. *"H-Correction"*, INRIA Report No. 3199, INRIA, pp.29, June 1997

[20] Do Carmo M. P. *Riemannian Geometry*, Birkhauser, Mathematics : Theory and Application, 1992

[21] Tam A.; Ait-Ali-Yahia D.; Robichaud M.P.; Moore M.; Kozel V.; Habashi W.G *Anisotropic mesh adaptation for 3D flows on structured and unstructured grids*, Computer Methods in Applied Mechanics and Engineering, Volume 189, Number 4, 29 September 2000, pp. 1205-1230(26)

[22] Lohner, R., and Beum, J. *Adaptive h-refinement on 3D unstructured grids for transient problems.* Int. Numer. Meth. Fluids 14 (1992), 1407-1419

[23] Lohner R., *Adaptative remeshing for transient problems*, Computer Methods in Applied Mechanics and Engineering, 1989, 75, pp. 195-214.

[24] Mavriplis, D. *Adaptive meshing techniques for viscous flow calculations on mixed-element unstructured meshes.* AIAA paper 97-0857 (1997)

[25] Dompierre, J., Vallet, M., Fortin, M., Bourgault, Y., and Habashi, W. *Anisotropic mesh adaptation: towards a solver and user independent cfd.* AIAA paper 97-0861 (1997)

[26] Gruau, C., and Coupez, T. *3D terahedral unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric.* Comput. Meth. Appl. Mech. Engrg. 194, 48-49 (2005), 4951-4976.

[27] Li, X., Shephard, M., and Beall, M. *3D anisotropic mesh adaptation by mesh modification.* Comput. Meth. Appl. Mech. Engrg. 194, 48-49 (2005), 4915-4950.

[28] Huang, W. *Metric tensors for anisotropic mesh generation.* J. Comput. Phys. 204 (2005), 633-665.

[29] J. Ruge and K. Stuben *Algebraic Multigrid Methods*: in Multigrid Methods (S. Mc Cormick Ed), Frontiers in Applied Math, Vol 5, SIAM, Philadelphia.

[30] P. Vanek, J. Mandel, and M. Brezina, *Algebraic Multigrid by Smoothed Aggregation for Second and Fourth Order Elliptic Problems*, Computing 56 (1996) pp 179-196.

[31] A. Janka, H. Guillard and P. Vanek *Convergence of Algebraic Multigrid based on smoothed aggregation II : Extension to a Petrov-Galerkin Method* INRIA technical report, No 3683, 1999

[32] H. Guillard and P. Vanek *An Aggregation Multigrid Solver for Convection-Diffusion Problems on Unstructured Meshes* University of Colorado in Denver, Center for Computational Mathematics, Report UCD-CCM No 130

[33] G. Carre, G. Carte, H. Guillard and S. Lanteri *Multigrid strategies for CFD problems on unstructured meshes* in "Multigrid Methods VI", Krys Riemslagh (ed), Lecture Notes in Computational Science and Engineering, 14, p 1–11, Springer.

[34] Habashi W.G., Dompierre J., Bourgault Y., Ait-Ali-Yahia D., Fortin M. and Vallet M.-G., *Anisotropic Mesh Adaptation: Towards User-Independent, Mesh-Independent and Solver-Independent CFD Solutions: Part I: General Principles*, International Journal for Numerical Methods in Fluids, 2000, 32, pp. 725-744.

[35] Tchon K.-F., Khachan Y., Guibault F. and Camarero R., *Constucting anisotropic geometric metrics using octree and skeletons*, pp. 293-304, 2003, Santa Fe, 12th Int. Meshing Roundtable.

[36] Peraire J. and Morgan K., *Unstructured mesh generation including directional refinement for aerodynamics flow simulation*, Finite Elements in Analysis and Design, 1997, 25, pp. 343-355.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399